Contents

30. Score Sets and Kings **1501**
30.1. Score sets in 1-tournaments 1502
30.1.1. Determining the score set 1502
30.1.2. Tournaments with prescribed score set 1504
30.2. Score sets in oriented graphs 1511
30.2.1. Oriented graphs with prescribed scoresets 1513
30.3. Unicity of score sets 1518
30.3.1. 1-unique score sets 1518
30.3.2. 2-unique score sets 1520
30.4. Kings and serfs in tournaments 1522
30.5. Weak kings in oriented graphs 1530
Bibliography **1542**
Subject Index **1545**
Name Index **1547**

30. Score Sets and Kings

The idea of comparison-based ranking has been discussed earlier in the chapter *Comparison based ranking*, where score sequence was introduced as a way of ranking vertices in a tournament. Oriented graphs are generalizations of tournaments. In fact, just like one can think of a tournament as expressing the results of a round-robin competition without ties (with vertices representing players and arrows pointing to the defeated players), one can think of an oriented graph as a round-robin competition with ties allowed (ties are represented by not drawing the corresponding arcs). Figure 30.1 shows the results of a round-robin competition involving 4 players a, b, c

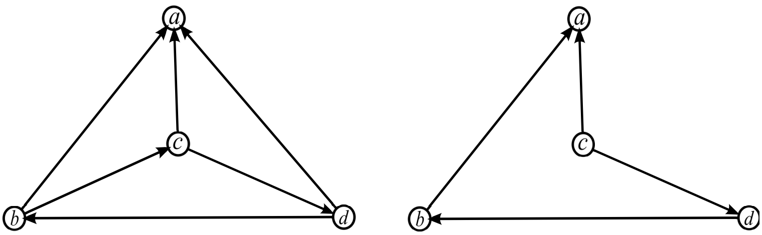

Figure 30.1 A round-robin competition involving 4 players.

and d, with (a) ties not allowed and (b) ties allowed. In the first instance there is always a winner and a loser whenever two players square off, while in the latter case player a ties with player d and player b ties with player c.

In 2009 Antal Iványi studied directed graphs, in which every pair of different vertices is connected with at least a and at most b arcs. He named them (a, b, n)-tournaments or simply (a, b)-tournament.

If $a = b = k$, then the (a, b)-tournaments are called k-tournaments. In this chapter we deal first of all with 1-tournaments and $(0, 1)$-tournaments. $(0, 1)$-tournaments are in some sense equivalent with $(2, 2)$-tournaments. We use the simple notations 1-tournament T_n^1, 2-tournament T_n^2, ..., k-tournament T_n^k, ... It is worth mentioning that T_n^1 is a classical tournament, while oriented graphs are $(0, 1)$-tournaments. If we allow loops then every directed graph is some (a, b, n)-tournament (see the Chapter **??** (*Comparison Based Ranking*) of this book).

We discuss two concepts related with (a, b)-tournaments, namely score sets and kings. A *score set* is just the set of different scores (out-degrees) of vertices, while a *king* is a dominant vertex. We shall study both concepts for 1-tournaments first and then extend these to the more general setting of oriented graphs.

Although we present algorithms for finding score sets and kings in 1-tournaments and $(0,1)$-tournaments, much of the focus is on constructing tournaments with special properties such as having a prescribed score set or a fixed number of kings. Since players in a tournament are represented by vertices, we shall use the words player and vertex interchangeably throughout this chapter without affecting the meaning.

We adopt the standard notation $T(V, A)$ to denote a tournament with vertex set V and arc set A. We denote the number of vertices by n, and the out-degree matrix by $\mathcal{M}$, and the in-degree matrix by N. Furthermore, we use the term n-tournament and the notation T_n^k to represent a tournament with n vertices and exactly k arcs between the elements of any pair of different vertices. In a similar way R_n^k and N_n denote a regular, resp. a null graph. When there is no ambiguity we omit one or even both indices shall refer to the corresponding tournaments as T, R. and N.

In Section 30.1 the score sets of 1-tournaments are discussed, while Section 30.2 deals with the sore sets of oriented graphs. In Section 30.3 the conditions of the unique reconstruction of the score sets are considered at first for k-tournaments, then in more details for 1-tournaments and 2-tournaments. In Section 30.4 and Section 30.5 results connected with different kings of tournaments are presented.

Some long and accessible proofs are omitted. In these cases the Reader can find the coordinates of the proof in *Chapter notes* and *Bibliography*.

30.1. Score sets in 1-tournaments

In a round-robin competition with no ties allowed, what are the sets of nonnegative integers that can arise as scores of players? Note that here we are not interested in the scores of individual players (the score sequence), rather we are looking for the sets of nonnegative integers with each integer being the score of at least one player in the tournament. This question motivates the study of *score sets* of tournaments.

The set of different scores of vertices of a tournament is called the ***score set*** of the tournament. In other words, the score set is actually the score sequence of a tournament with repetitions removed. For example the tournament given in Figure 30.2 has score sequence $[0, 2, 2, 2]$, whereas the score set of this tournament is $\{0, 2\}$. Figure 30.3 shows the out-degree matrix of the tournament represented on Figure 30.2.

30.1.1. Determining the score set

Determining the score set of a tournament $T(V, A)$ is quite easy. The following algorithm Set1 takes the data of a tournament $T(V, A)$ as input and returns the score set $\mathcal{S}$ of T.

The procedures of this chapter are written according to the third edition of the textbook *Introduction to Algorithms* published by T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein in 2009.

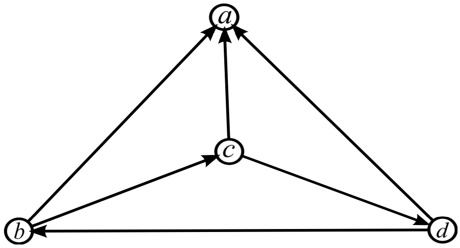

Figure 30.2 A tournament with score set $\{0, 2\}$.

vertex/vertex	a	b	c	d	Score
a	—	0	0	0	0
b	1	—	1	0	2
c	1	0	—	1	2
d	1	1	0	—	2

Figure 30.3 Out-degree matrix of the tournament represented in Figure 30.2.

Set1(n, V, A)

```
1  S = ∅
2  for all vertex u ∈ V
3       s = 0
4       for all vertex v ∈ V
5            if (u, v) ∈ A                    // is (u, v) an arc of T?
6                 s = s + 1
7       if s ∉ S                              // is the found score new?
8            S = S ∪ {s}
9  return S
```

Since the scores of the vertices depend on $n(n-1)$ out-degrees, any algorithm determining the score set requires $\Omega(n^2)$ time. Due to the embedded loops in lines 02–08 the running time of Set1 is $\Omega(n^2)$ even in the best case. The precise order of the running time depends among others on the implementation of the **if** instruction in line 07. E.g., if line 07 is implemented by the comparison of the actual score with the elements of S, then the running time is $\Theta(n^3)$ for a score sequence containing different elements and is $\Theta(n^2)$ for a regular tournament.

Out-degree matrix $\mathcal{M}_{n\times n} = [m_{ij}]_{n\times n}$ is a useful tool in the implementation of graph algorithms. The input of the following algorithm Quick-Set1 is n and $\mathcal{M}$, and the output is the score sequence $\mathbf{s}$ as a nonincreasingly ordered sequence and the score set $\mathcal{S}$ as an increasingly ordered sequence. Quick-Set1 calls the well-known sorting procedure Insertion-Sort.

Quick-Set1($n, \mathcal{M}$)

1 $\mathcal{S} = \emptyset$
2 **for** $i = 1$ **to** n
3 $\quad s_i = 0$
4 $\quad$ **for** $j = 1$ **to** n
5 $\quad\quad s_i = s_i + m_{ij}$ // score sequence is computed
6 $S_1 = s_1$
7 Insertion-Sort($\mathbf{s}$) **if** $s \notin \mathcal{S}$ // sorting of the score vector
8 **for** $i = 2$ **to** n
9 $\quad$ **if** $s_i \neq s_{i-1}$
10 $\quad\quad S_k = s_i$
11 $\quad\quad k = k + 1$
12 **return** $\mathbf{s}$, $\mathcal{S}$

Since the embedded loops in lines 02–05 need $\Theta(n^2)$ time, and the remaining part of the code requires less, the running time of Quick-Set1 is $\Theta(n^2)$ in all cases.

30.1.2. Tournaments with prescribed score set

Constructing a tournament with a prescribed score set is more difficult than determining the score set. Quite surprisingly, if sufficiently many players participate in a tournament then any finite set of nonnegative integers can arise as a score set. This was conjectured by K. B. Reid in 1978 and turned out to be a relatively challenging problem.

Reid proved the result when $| \mathcal{S} | = 1, 2$ or 3, or if $\mathcal{S}$ contains consecutive terms of an arithmetic or geometric progression. That is, Reid showed that any set of one, two or three nonnegative integers is a score set of some tournament and additionally, any set of the form $\{s, s+d, s+2d, \ldots, s+pd\}$ for $s > 0, d > 1$ or $\{s, sd, sd^2, \ldots, sd^p\}$ for $s \geq 0, d > 0$, is a score set of some tournament. Hager settled the cases $|\mathcal{S}| = 4$ and $|\mathcal{S}| = 5$ in 1986 and finally in 1987, T. Yao gave an existence proof of the general Reid's conjecture based on arithmetic analysis.

Theorem 30.1 (Yao, 1988) *Every finite nonempty set $\mathcal{S}$ of nonnegative integers is the score set of some tournament.*

Let us try to formulate Reid's conjecture purely as a statement about numbers. Let $\mathcal{S} = \{s_1, \ldots, s_p\}$ be an increasing sequence of nonnegative integers. The conjecture means that there exist positive integers $x_1, \ldots, x_p$ such that

$$\mathcal{S} = (s_1^{x_1}, \ldots, s_2^{x_2} \ldots, s_p^{x_p})$$

is the score sequence of some 1-tournament with $\sum_{i=1}^{p} x_i = n$ vertices. By Landau's theorem, $\mathbf{a} = (a_1, \ldots, a_n)$, with $a_1 \leq \cdots \leq a_n$, is the score sequence of some 1-tournament T_n if and only if $\sum_{i=1}^{k} a_i \geq \binom{k}{2}$, for $k = 1, \ldots, n-1$ and $\sum_{i=1}^{n} a_i = \binom{n}{2}$. Thus it can be readily seen that Reid's conjecture is equivalent to the following statement.

For every nonempty set of nonnegative integers $S = \{s_1, \ldots, s_p\}$, where $s_1 <$

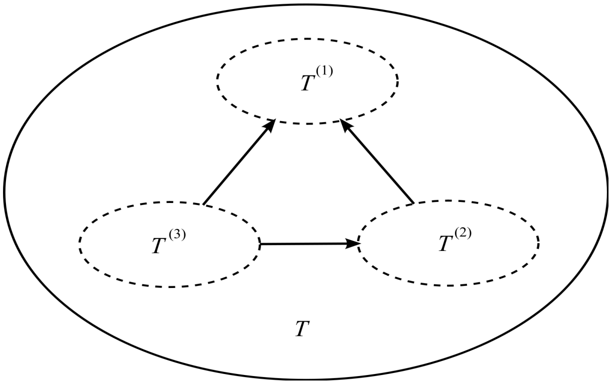

Figure 30.4 Construction of tournament T with odd number of distinct scores.

$\cdots < s_p$, there exist positive integers $x_1, \ldots, x_p$, such that

$$\sum_{i=1}^{k} s_i x_i \geq \binom{\sum_{i=1}^{k} x_i}{2}, \quad \text{for } k = 1, \ldots, p-1\,, \tag{30.1}$$

$$\sum_{i=1}^{p} s_i x_i = \binom{\sum_{i=1}^{p} x_i}{2}\,. \tag{30.2}$$

It is this equivalent formulation of Reid's conjecture that led to Yao's proof. The proof is not combinatorial in nature, but uses first of all some results of number theory. Commenting on Yao's proof Qiao Li wrote in 2006 in the *Annals of New York Academy of Sciences:*

> *Yao's proof is the first proof of the conjecture, but I do not think it is the last one. I hope a shorter and simpler new proof will be coming in the near future.*

However, the prophecized constructive proof has not been discovered yet. This is in sharp contrast with Landau's theorem on score sequences, for which several proofs have emerged over the years. Recently, S. Pirzada and T. A. Naikoo gave a constructive combinatorial proof of a new special case of Reid's theorem. Their proof gives an algorithm for constructing a tournament with the prescribed score set, provided the score increments are increasing.

Theorem 30.2 (Pirzada and Naikoo, 2008) *If $a_1, a_2, \ldots, a_p$ are nonnegative integers with $a_1 < a_2 < \cdots < a_p$, then there exists a 1-tournament T with score set*

$$S = \left\{ s_1 = a_1, s_2 = \sum_{i=1}^{2} a_i, \ldots, s_p = \sum_{i=1}^{p} a_i \right\}\,. \tag{30.3}$$

Since any set of nonnegative integers can be written in the form of 30.3, the above theorem is applicable to all sets of nonnegative integers $S = \{s_1, s_2, \ldots, s_p\}$

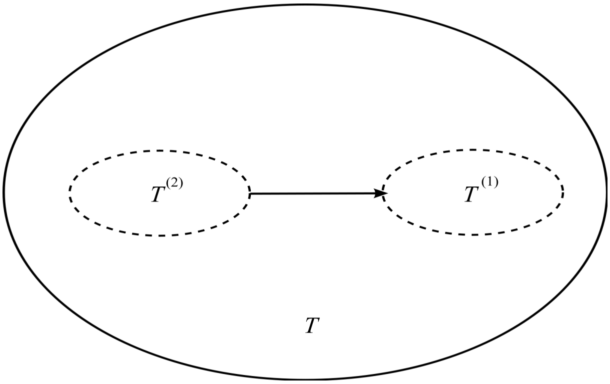

Figure 30.5 Construction of tournament T with even number of distinct scores.

with increasing increments (i.e., $s_1 < s_2 - s_1 < s_3 - s_2 < \cdots < s_p - s_{p-1}$.) The importance of Pirzada-Naikoo proof of Theorem 30.2 is augmented by the fact that Yao's original proof is not constructive and is not accessible to a broad audience[1].

The following recursive algorithm is based on Pirzada and Naikoo's proof of Theorem 30.2. The algorithm takes the set of increments $I_p = \{a_1 < a_2 < \cdots < a_p\}$ of the score set S as input and returns a tournament T whose score set is S. Let $X_t = \{a_1 < a_2 < \cdots < a_t\}$ for $1 \le t \le p$. Let R_n denote the regular tournament on n vertices and let $T^{(1)} \oplus T^{(2)}$ denote the vertex and arc disjoint union of tournaments $T^{(1)}$ and $T^{(2)}$.

Score-Reconstruction1(p, I_p)

```
if    p is odd
      print Odd(p, I_p)
else print Even(p, I_p)
```

This algorithm calls one of the two following recursive procedures ODD and Even according to the parity of p. The input of both algorithm is some prefix X_t of the sequence of the increments $a_1, a_2, \ldots, a_t$, and the output is a tournament having the score set corresponding to the given increments.

Odd(t, X_t)

```
if    t == 1
      return R_{2a_1+1}
```

[1] Yao's proof originally appeared in Chinese in the journal *Kexue Tongbao*. Later in 1989, the proof was published in English in the *Chinese Science Bulletin*. Unfortunately neither are accessible through the world wide web, although the English version is available to subscribers of the *Chinese Science Bulletin*. In Hungary this journal is accessible in the Library of Technical and Economical University of Budapest.

vertex/vertex	v_1	v_2	v_3	v_4	v_5	Score
v_1	—	1	1	0	0	2
v_2	0	—	1	1	0	2
v_3	0	0	—	1	1	2
v_4	1	0	0	—	1	2
v_5	1	1	0	—	0	2

Figure 30.6 Out-degree matrix of the tournament $T_5^{(3)}$.

3 **else** $T_t^{(3)} = R_{(2(a_t - a_{t-1} + a_{t-2} - a_{t-3} + \cdots + a_3 - a_2 + a_1) + 1)}$
4 $\quad T_t^{(2)} = R_{2(a_{t-1} - a_{t-2} + a_{t-3} - a_{t-2} + \cdots + a_4 - a_3 + a_2 - a_1 - 1) + 1}$
5 $\quad t = t - 2$
6 $\quad T_t^{(1)} = \text{ODD}(t, X_t)$
7 $\quad T_t = T_t^{(3)} \oplus T_t^{(2)} \oplus T_t^{(1)}$
8 $\quad T_t = T +$ arcs such that
9 $\qquad\quad T_t^{(2)}$ dominates $T_t^{(1)}$
10 $\qquad\quad T_t^{(3)}$ dominates $T_t^{(1)}$
11 $\qquad\quad T_t^{(3)}$ dominates $T_t^{(2)}$
12 **return** T_t

We can remark that the tournament constructed by the first execution of line 03 of Odd contains the vertices whose score is a_p, while the tournament constructed in line 04 contains the vertices whose score is a_{p-1} in the tournament appearing as output. The vertices having smaller scores appear during the later execution of lines 03 and 04 with exception of the vertices having score a_1 since those vertices will be added to the output in line 02.

Even(t, X_t)

1 $\quad T_t^{(2)} = R_{2(a_t - a_{t-1} + a_{t-2} - a_{t-3} + \cdots + a_4 - a_3 + a_2 - a_1 - 1) + 1}$
2 $\quad t = t - 1$
3 $\quad T_t^{(1)} = \text{ODD}(t, X_t)$
4 $\quad T_t = T_t^{(2)} \oplus T_t^{(1)}$
5 $\quad T_t = T +$ arcs such that $T_t^{(2)}$ dominates $T_t^{(1)}$
6 **return** T_t

Since the algorithm is complicated, let's consider an example.

Example 30.1 Let $p = 5$ and $I_5 = \{0, 1, 2, 3, 4\}$. Since p is odd, Score-Reconstruction1 calls Odd in line 02 with parameters 5 and I_5.

The first step of Odd is the construction of $T_5^{(3)} = T_{2(4-3+2-1+0)+1} = T_5$ in line 03. Denoting the vertices of this regular 5-tournament by v_1, v_2, v_3, v_4, v_5 and using the result of Exercise 30.1-1 we get the out-degree matrix shown in Figure 30.6.

vertex/vertex	v_9	v_{10}	v_{11}	v_{12}	v_{13}	Score
v_9	—	1	0	1	1	3
v_{10}	0	—	1	1	1	3
v_{11}	1	0	—	1	1	3
v_{12}	0	0	0	—	1	1
v_{13}	0	0	0	0	—	0

Figure 30.7 Out-degree matrix of the tournament $T_5^{(3)}$.

v/v	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}	v_{11}	v_{12}	v_{13}	Score
v_1	—	0	0	0	0	0	0	0	0	0	0	0	0	0
v_2	1	—	0	0	0	0	0	0	0	0	0	0	0	1
v_3	1	1	—	1	0	0	0	0	0	0	0	0	0	3
v_4	1	1	0	—	1	0	0	0	0	0	0	0	0	3
v_5	1	1	1	0	—	0	0	0	0	0	0	0	0	3
v_6	1	1	1	1	1	—	1	0	0	0	0	0	0	6
v_7	1	1	1	1	1	0	—	1	0	0	0	0	0	6
v_8	1	1	1	1	1	1	0	—	0	0	0	0	0	6
v_9	1	1	1	1	1	1	1	1	—	1	0	1	1	10
v_{10}	1	1	1	1	1	1	1	1	0	—	1	1	0	10
v_{11}	1	1	1	1	1	1	1	1	0	0	—	1	1	10
v_{12}	1	1	1	1	1	1	1	1	1	0	0	—	1	10
v_{13}	1	1	1	1	1	1	1	1	1	1	0	0	—	10

Figure 30.8 Out-degree matrix of the tournament T_5.

The second step of ODD is the construction of $T_5^{(2)} = T_{2(3-2+1-0-1)+1} = T_3$. Let v_6, v_7 and v_8 be the vertices of this tournament.

The third step of ODD is the recursive call with parameters $p = 3$ and $X_3 = \{2,\ 1,\ 0\}$. The fourth action of ODD is the construction of $T_3^{(3)} = T_{2(2-1+0)+1} = T_3$. Let v_9, v_{10} and v_{11} be the vertices of this tournament. The fifth step is the construction of $T_3^{(2)} = T_{2(2-1+0-1)+1} = T_1$. Let v_{12} be the only vertex of this graph. The sixth action is the call of ODD with parameters $t = 1$ and $X_1 = \{0\}$. Now the number of increments equals to 1, therefore the algorithm constructs $T_1^{(1)} = T_1$ in line 02.

The seventh step is the construction of T in line 07, then the eighth step is adding new arcs (according to lines 08–11) to the actual T constructed in line 07 and consisting from 3 regular tournaments having altogether 5 vertices (v_{13}, v_{12}, v_{11}, v_{10}, v_9). The result is shown in Figure 30.7.

Ninth step of ODD is joining the tournaments T_5 and T_3 to T and the final step is adding of the domination arcs. The out-degree matrix of the output T_5 of ODD is shown in Figure 30.8.

Correctness of the algorithm

Let $I = \{a_1, a_2, \ldots, a_p\}$ be a set of p nonnegative integers with $a_1 < a_2 < \cdots < a_p$.

Score-Reconstruction1 performs two types of recursions: first if p is odd and the second if p is even. Assume p to be odd. For $p = 1$, the set I contains one nonnegative integer a_1 and the algorithm returns the regular tournament T_{2a_1+1} as output. Note that each vertex of T_{2a_1+1} has score $\left(\frac{2a_1+1-1}{2}\right) = a_1$, so that score set of T_{2a_1+1} is $S = \{s_1 = a_1\}$. This shows that the algorithm is correct for $p = 1$.

If $p = 3$, then the set of increments I consists of three nonnegative integers $\{a_1, a_2, a_3\}$ with $a_1 < a_2 < a_3$. Now $a_3 > a_2$, therefore $a_3 - a_2 > 0$, so that $a_3 - a_2 + a_1 > 0$ as $a_1 \geq 0$. Let $T^{(3)}$ be a regular tournament having $2(a_3 - a_2 + a_1) + 1$ vertices. Then each vertex of $T^{(3)}$ has score $\left(\frac{2(a_3-a_2+a_1)+1-1}{2}\right) = a_3 - a_2 + a_1$.

Again, since $a_2 > a_1$, therefore $a_2 - a_1 > 0$, so that $a_2 - a_1 - 1 \geq 0$. Let $T^{(2)}$ be a regular tournament having $2(a_2 - a_1 - 1) + 1$ vertices. Then each vertex of $T^{(2)}$ has score $\left(\frac{2(a_2-a_1-1)+1-1}{2}\right) = a_2 - a_1 - 1$. Also since $a_1 \geq 0$, let $T^{(1)}$ be a regular tournament having $2a_1+1$ vertices. Then each vertex of T_1 has score $\left(\frac{2a_1+1-1}{2}\right) = a_1$.

If $p = 3$, Score-Reconstruction1 outputs a tournament T whose vertex set is the disjoint union of vertex sets of $T^{(1)}$, $T^{(2)}$ and $T^{(3)}$ and whose arc set contains all the arcs of $T^{(1)}$, $T^{(2)}$ and $T^{(3)}$ such that every vertex of $T^{(2)}$ dominates each vertex of $T^{(1)}$, and every vertex of $T^{(3)}$ dominates each vertex of $T^{(1)}$ and $T^{(2)}$. Thus T has $2a_1 + 1 + 2(a_2 - a_1 - 1) + 1 + 2(a_3 - a_2 + a_1) + 1 = 2(a_1 + a_3) + 1$ vertices with score set

$$
\begin{aligned}
S &= \{a_1, a_2 - a_1 - 1 + 2a_1 + 1, a_3 - a_2 + a_1 + 2(a_2 - a_1 - 1) + 1 + 2a_1 + 1\} \\
&= \left\{a_1, \sum_{i=1}^{2} a_i, \sum_{i=1}^{3} a_i\right\} .
\end{aligned}
$$

This shows that the algorithm is correct for $p = 3$ too. When the set I of increments consists of an odd number of nonnegative integers, the algorithm recursively builds the required tournament by using the procedure Odd. To see this, assume that the algorithm works for all odd numbers upto p. That is, if $a_1, a_2, \ldots, a_p$ are p nonnegative integers with $a_1 < a_2 < \cdots < a_p$, then the algorithm outputs a tournament having $2(a_1 + a_3 + \ldots + a_p) + 1$ vertices with score set $\{a_1, \sum_{i=1}^{2} a_i, \ldots, \sum_{i=1}^{p} a_i\}$. Let us call this tournament $T^{(1)}$.

We now show how the algorithm constructs a tournament with $p+2$ vertices with score set $\{a_1, \sum_{i=1}^{2} a_i, \ldots, \sum_{i=1}^{p+2} a_i\}$, where $a_1, a_2, \ldots, a_{p+2}$ are $p + 2$ nonnegative integers with $a_1 < a_2 < \cdots < a_{p+2}$.

Since $a_2 > a_1, a_4 > a_3, \ldots, a_{p-1} > a_{p-2}, a_{p+1} > a_p$. therefore $a_2 - a_1 > 0$, $a_4 - a_3 > 0, \ldots, a_{p-1} - a_{p-2} > 0$, $a_{p+1} - a_p > 0$, so that $a_{p+1} - a_p + a_{p-1} - a_{p-2} + \ldots + a_4 - a_3 + a_2 - a_1 > 0$, that is, $a_{p+1} - a_p + a_p - 1 - a_{p-2} + \ldots + a_4 - a_3 + a_2 - a_1 - 1 \geq 0$.

The procedure Odd constructs $T^{(2)}$ as a regular tournament having $2(a_{p+1} - a_p + a_{p-1} - a_{p-2} + \cdots + a_4 - a_3 + a_2 - a_1 - 1) + 1$ vertices. Each vertex of $T^{(2)}$ has score

$$
\begin{aligned}
&\frac{2(a_{p+1} - a_p + a_{p-1} - a_{p-2} + \ldots + a_4 - a_3 + a_2 - a_1 - 1) + 1 - 1}{2} \\
&= a_{p+1} - a_p + a_{p-1} - a_{p-2} + \cdots + a_4 - a_3 + a_2 - a_1 - 1 \, .
\end{aligned}
$$

Again, $a_3 > a_2, \ldots, a_p > a_{p-1}, a_{p+2} > a_{p+1}$, therefore $a_3 - a_2 > 0, \ldots, a_p -$

$a_{p-1} > 0, a_{p+2} - a_{p+1} > 0$, so that $a_{p+2} - a_{p+1} + a_p - a_{p-1} + \cdots + a_3 - a_2 + a_1 > 0$ as $a_1 \geq 0$.

The procedure ODD constructs $T^{(3)}$ as a regular tournament having $2(a_{p+2} - a_{p+1} + a_p - a_{p-1} + \cdots + a_3 - a_2 + a_1) + 1$ vertices. Each vertex of $T^{(3)}$ has score

$$\begin{aligned} &\frac{2(a_{p+2} - a_{p+1} + a_p - a_{p-1} + \cdots + a_3 - a_2 + a_1) + 1 - 1}{2} \\ &= a_{p+2} - a_{p+1} + a_p - a_{p-1} + \cdots + a_3 - a_2 + a_1 \,. \end{aligned}$$

Now SCORE-RECONSTRUCTION1 sets $T = T^{(1)} \oplus T^{(2)} \oplus T^{(3)}$ and adds additional arcs in such a way that every vertex of $T^{(2)}$ dominates each vertex of $T^{(1)}$, and every vertex of $T^{(3)}$ dominates each vertex of $T^{(1)}$ and $T^{(2)}$. Therefore T is a tournament having

$$\begin{aligned} &2(a_1 + a_3 + \cdots + a_p) + 1 + 2(a_{p+1}a_p + a_{p1}a_{p2} + \cdots + a_4a_3 + a_2a_1) + 1 \\ &\qquad\qquad +2(a_{p+2}a_{p+1} + a_pa_{p-1} + \cdots + a_3a_2 + a_1) + 1 \\ &= 2(a_1 + a_3 + \cdots + a_{p+2}) + 1 \end{aligned}$$

vertices with score set

$$S = \left\{ a_1, \sum_{i=1}^{2} a_i, \ldots, \sum_{i=1}^{p} a_i, \sum_{i=1}^{p+1} a_i, \sum_{i=1}^{p+2} a_i \right\} .$$

Hence by induction, the algorithm is correct for all odd p.

To prove the correctness for even case, note that if p is odd, then $p+1$ is even. Let $a_1, a_2, \ldots, a_{p+1}$ be $p+1$ nonnegative integers with $a_1 < a_2 < \cdots < a_{p+1}$.. Therefore $a_1 < a_2 < \cdots < a_p$, where p is odd. The procedure EVEN uses the procedure ODD to generate a tournament $T^{(1)}$ having $2(a_1 + a_3 + \cdots + a_p) + 1$ vertices with score set $S = \{a_1, \sum_{i=1}^{2} a_i, \ldots, \sum_{i=1}^{p} a_i\}$.

Also, since $a_2 > a_1$, $a_4 > a_3, \ldots, a_{p-1} > a_{p-2}, a_{p+1} > a_p$, the procedure EVEN generates a regular tournament $T^{(2)}$ having $2(a_{p+1} - a_p + a_{p-1} - a_{p-2} + \cdots + a_4 - a_3 + a_2 - a_1 - 1) + 1$ vertices such that the score for each vertex is $a_{p+1} - a_p + a_{p-1} - a_{p-2} + \cdots + a_4 - a_3 + a_2 - a_1 - 1$.

Finally the algorithm generates the tournament $T^{(1)} \oplus T^{(2)}$ and adds additional arcs so that every vertex of $T^{(2)}$ dominates each vertex of $T^{(1)}$. The resulting tournament T consists of

$$\begin{aligned} &2(a_1 + a_3 + \cdots + a_{p-2} + a_p) + 1 \\ &\qquad +2(a_{p+1} - a_p + a_{p-1} - a_{p-2} + \cdots + a_4 - a_3 + a_2 - a_1 - 1) + 1 \\ &= 2(a_2 + a_4 + \cdots + a_{p+1}) \end{aligned}$$

vertices and has score set

$$
\begin{aligned}
S = \{a_1, \sum_{i=1}^{2} a_i, \ldots, \sum_{i=1}^{p} a_i, \\
& a_{p+1} - a_p + a_{p-1} - a_{p-2} + \cdots + a_4 - a_3 + a_2 - a_1 - 1 \\
& \qquad +2(a_1 + a_3 + \cdots + a_{p-2} + a_p) + 1\} \\
= \{a_1, \sum_{i=1}^{2} a_i, \ldots, \sum_{i=1}^{p+1} a_i\}\,.
\end{aligned}
$$

This shows that the algorithm is correct for even p as well.

Computational complexity

The running time of Score-Reconstruction1 depends on the size of the score set $|S|$ as well as the largest increment $a_p = s_p - s_{p-1}$. The details are left as a problem for the Reader (see Exercise 30.1-1).

Exercises

30.1-1 The out-degree matrix $\mathcal{M}$ of a tournament is defined as a $0-1$ matrix with (i,j) entry equal to 1 if player v_i defeats player v_j and 0 otherwise (see (30.13)). A tournament is completely determined by its out-degree matrix. Write an $O(n^2)$ algorithm to generate the out-degree matrix of a regular tournament on n vertices, where n is any odd positive integer. *Hint.* Circularly place $\binom{n-1}{2}$ ones in each row.

30.1-2 Use Exercise 30.1-1 and the discussion in this section to determine the worst-case running time of Score-Reconstruction1.

30.1-3 Obtain the out-degree matrix of a tournament with score set $\{1,3,6\}$. How many vertices does this tournament have? Draw this tournament and give its outdegree-matrix.

30.1-4 Use the tournament obtained in Exercise 30.1-3 to generate the out-degree matrix of a 1-tournament with score set $\{1,3,6,10\}$. Write the score sequence of your tournament.

30.2. Score sets in oriented graphs

Oriented graphs are generalizations of tournaments. Formally, an ***oriented graph*** $D(V,A)$ with vertex set V and arc set A is a digraph with no symmetric pairs of directed arcs and without loops. In other words oriented graph is a directed graph in which every pair of different vertices is connected with at most one arc, or oriented graphs are $(0,1)$-*tournaments.*

Figure 30.9 shows an oriented graph with score sequence $[1,3,3,5]$ and the coressponding score set $\{1,3,5\}$.

Thus tournaments are complete oriented graphs, in the sense that any pair of vertices in a tournament is joined exactly by one arc. Several concepts defined for tournaments can be extended in a meaningful way to oriented graphs. For example score of a player (vertex) in a tournament is defined as its out-degree, as a player

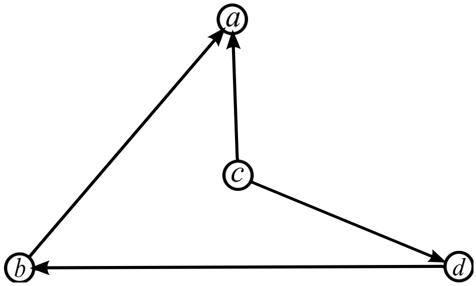

Figure 30.9 An oriented graph with score sequence $[1, 3, 3, 5]$ and score set $\{1, 3, 5\}$.

either wins (and earns one point) or looses (earning no points) a two-way clash. In 1991, Peter Avery introduced the score structure for oriented graphs based on the intuition that in a round-robin competition with ties allowed, a player may earn two, one or no points in case the player wins, looses or makes a tie respectively.

More precisely, the score of a vertex v_i in a k-tournament D with n vertices is defined as

$$a(v_i) = a_i = n - 1 + d^+_{v_i} - d^-_{v_i} ,$$

where $d^+_{v_i}$ and $d^-_{v_i}$ are the out-degree and in-degree, respectively, of v_i. The score sequence of an oriented graph is formed by listing the vertex scores in non-decreasing order. If we denote the number of non-arcs in D containing the vertex v_i as $d^*_{v_i}$, then

$$a_i = 2d^+_{v_i} + d^*_{v_i} .$$

With this score structure, an oriented graph can be interpreted as the result of a round-robin competition in which ties (draws) are allowed, that is, the players play each other once, with an arc from player u to v if and only if u defeats v. A player receives two points for each win, and one point for each tie.

It is worth to remark that this is a sophisticated score structure comparing with the simple and natural structure of 2-tournaments.

Avery gave a complete characterization of score sequences of oriented graphs similar to Landau's theorem.

Theorem 30.3 (Avery, 1991) *A nondecreasing sequence $A = [a_1, \ldots, a_n]$ of non-negative integers is the score sequence of an oriented graph if and only if*

$$\sum_{i=1}^{k} a_i \geq k(k-1) \tag{30.4}$$

for $1 \leq k \leq n$ with equality when $k = n$.

Proof This theorem is a special case of the theorem proved by Moon in 1963 or the theorem proved by Kemnitz and Dulff in 1997 (see the theorem and its proof in Chapter **??**, that is chapter *Comparison Based Ranking*). ■

Just as in the case of 1-tournaments, the ***score set of an oriented graph*** is defined as the set of scores of its vertices. It is worth noting that a $(0,1)$-tournament has different score sets under Avery's and Landau's score structures. In fact, the score of a vertex v under Avery's score structure is twice the score of v under Landau's score structure. This is obviously due to Avery's assumption that a win contributes 2 points to the score.

The score set of an oriented graph can be determined by adapting Quick-Set2 as follows:

```
Quick-Set2(n, M)
1  S = ∅
2  for i = 1 to n
3      s_i = 0
4      for j = 1 to n
5          s_i = s_i + 2m_ij
6          if m_ij==0 and m_ji == 0
7             s_i = s_i + 1                    // score sequence is computed
8  S_1 = s_1
9  k = 2
10 for i = 2 to n
11     if s_i ≠ s_{i-1}                        // is the found score new?
12        S_k = s_i
13        k = k + 1
14 return s, S
```

The running time of Quick-Set2 is $\Theta(n^2)$ since the nested loop in lines 02–07 requires $\Theta(n^2)$ the remaining lines require $\Theta(n)$ time.

30.2.1. Oriented graphs with prescribed scoresets

In Section 30.2 we discussed score sets of tournaments and noted that every non-empty set of nonnegative integers is the score set of some tournament. In this section we study the corresponding question for oriented graphs, i.e., which sets of nonnegative integers can arise as score sets of oriented graphs. Pirzada and Naikoo investigated this question and gave two sufficient conditions for a set of nonnegative integers to be the score set of some oriented graph.

Theorem 30.4 (Pirzada, Naikoo, 2008) *Let a, d, n nonnegative integers, $a > 0$and $S = \{a, ad, ad^2, \ldots, ad^n\}$, with $d > 2$ or $d = 2$ and $n > 1$. Then there exists an oriented graph with score set S except for $a = 1,\ d = 2,\ n > 0$ and for $a = 1,\ d = 3,\ n > 0$.*

Theorem 30.5 (Pirzada, Naikoo, 2008) *If n is a positive integer and $a_1, a_2, \ldots, a_n$ are nonnegative integers with $a_1 < a_2 < \cdots < a_n$, then there exists an oriented graph with $a_n + 1$ vertices and with score set $S = \{s_1, s_2, \ldots, s_n\}$, where*

$$s_i = \begin{cases} s_{i-1} + a_i + 1 & \text{for } i > 1\,, \\ s_i & \text{for } i = 1\,. \end{cases} \tag{30.5}$$

Thus any set of positive integers whose elements form a geometric progression is the score set of some oriented graph with few exceptions and any set of nonnegative integers whose elements are of the form (30.5) is also a score set. It follows that every singleton set of nonnegative integers is the score set of some oriented graph. On the other hand, for any positive integer n, the sets $\{1, 2, 2^2, \ldots, 2^n\}$ and $\{1, 3, 3^2, \ldots, 3^n\}$ cannot be the score sets of an oriented graph. Therefore, unlike in the case of tournaments, not all sets of nonnegative integers are score sets of oriented graphs. So far no complete characterization of score sets of oriented graphs is known.

The proof of Theorem 30.4 depends on the following auxiliary assertion.

Lemma 30.6 (Naikoo, Pirzada, 2008) *The number of vertices in an oriented graph with at least two distinct scores does not exceed its largest score.*

Proof This assertion is the special case $k = 2$ of Lemma **??** due to Iványi and Phong. ■

Here we omit formal proofs of Theorems 30.4 and 30.5 since they can be found on the internet and since we will implicitly prove these theorems when we check the correctness of Geometric-Construction2 and Adding-Construction2, respectively.

We first present a recursive algorithm that takes positive integers a, d, and n, satisfying the condition of Theorem **??**, as input and generates a 2-tournament $D(V, A)$ with score set $\{a, ad, ad^2, \ldots, ad^n\}$. Let N_p denote the null digraph on p vertices, i.e., the digraph with n vertices and no arcs.

Geometric-Construction2(a, d, n)

```
if    n = 0
      D = N_{a+1}
      return D
if n = 1
   if a = 1 and d = 3
      X = N_{a+1}
      Y = N_{ad-a-1}
      D = X ⊕ Y
      add arcs to D such
      X dominates Y
      return D
   if a > 2 and d = 2
      X = N_2 with vertex set {u_1, u_2}
      Y = N_{a-2} with vertex set {v_1, v_2, ..., v_{a-2}}
      Z = N_a with vertex set {z_1, z_2, ..., z_a}
      D = X ⊕ Y ⊕ Z
      add arcs to D such that
      v_i → u_1 for i = 1, 2, ..., a
      v_i → u_2 for i = 1, 2, ..., a
      w_1 → u_1
```

$w_2 \to u_2$
$w_{i+2} \to v_i$ for $i = 1, 2, \ldots, a-2$
if $a = 1$ and $d = 3$
$X = N_{a+1}$
$Y = N_{ad-a-1}$
$D = X \oplus Y$
add arcs to D such
X dominates Y
return D

The running time of Geometric-Construction2 is ???????

Example 30.2 Let $a = 2,\ d = 2$ and $n = 2$. Then the prescribed score set is $\{2, 4, 8\}$. The first step is the call of Geometric with parameters $(2, 2, 2)$.

Algorithm description

If $n = 0$, then the algorithm returns the null digraph N_{a+1}. Note that N_{a+1} is well-defined as $a+1 > 0$. Each vertex of N_{a+1} has score $a+1-1+0-0 = a$. Therefore the score set of N_{a+1} is $S = \{a\}$. Thus the algorithm is correct for $n = 0$.

If $n = 1$, then four cases arise.

Case i). If $a = 1$ and $d > 3$, then $a+1 > 0$ and $ad - 2a - 1 > 0$. Construct an oriented graph D with vertex set $V = X \cup Y$, where $X \cap Y = \emptyset$, $|X| = a+1$, $|Y| = ad - 2a - 1$ and $Y \to X$. Then D has $ad - a$ vertices and the different scores are a and ad.

Case ii). If $a = 2$ and $d = 2$. Construct an oriented graph D with vertex set $\{v_1, v_2, v_3\}$ and v_4 and arc set $(v_1, v_3), (v_2, v_4)$. The score set of D is $\{2, 4\}$.

case iii). If $a > 2$ and $d = 2$, then with vertex set $V = X \cup Y$, where $X \cap Y = \emptyset$, $|X| = a+1$, $|Y| = ad - 2a - 1$ and $Y \to X$. Then the scores are a for all $x \in X$ and ad for all $y \in Y$.

Now we prove the correctness of Geometric by induction. That is, we show that if the algorithm is valid for $n = 0, 1, \ldots, p$ for some integer $p \geq 1$ then it is also valid for $n = p+1$. Let a and d be positive integers with $a > 0$ and $d > 1$ such that for $a = 1$, $d \neq 2, 3$. By the induction hypothesis the algorithm can construct an oriented graph $D^{(1)}$ with score set $\{a, ad, \ldots, ad^p\}$ and $a, ad, \ldots, ad^p$ are the distinct scores of the vertices of $D^{(1)}$. Let U be the vertex set of $D^{(1)}$.

There are three possibilities:

- $a = 1$ and $d > 3$,
- $a > 1$ and $d = 2$ or
- $a > 1$ and $d > 2$.

Obviously, for $d > 1$ in all the above cases we have $ad^{p+1} \geq 2ad^p$. Also the score set of $D^{(1)}$, namely $\{a, ad, \ldots, ad^p\}$, has at least two distinct scores for $p \geq 1$. Therefore, by Lemma 30.6 we have $|U| \leq ad^p$. Hence $ad^{p+1} \geq 2|U|$ so that $ad^{p+1} - 2|U| + 1 > 0$.

Let $N_{ad^{p+1}-2|U|+1}$ be the null digraph with vertex set X. The algorithm now generates the vertex and arc disjoint union $D = D^{(1)} \oplus N_{ad^{p+1}-2|U|+1}$ and adds an arc directed from each vertex in $N_{ad^{p+1}-2|U\ vert+1}$ to every vertex of $D^{(1)}$. The output $D(V, A)$ of GEOMETRIC-CONSTRUCTION2, therefore, has $|V| = |U| + ad^{p+1} - 2|U| + 1 = ad^{p+1} - |U| + 1$ vertices. Moreover, $a + |X| - |X| = a,$, $ad + |X| - |X| = ad$. $ad^2 + |X| - |X| = ad^2, \ldots, ad^p + |X| - |X| = ad^p$ are the distinct scores of the vertices in U, while $a_x = |U| - 1 + |V| - 0 = ad^{p+1} - |V| + 1 - 1 + |V| = ad^{p+1}$ for all vertices $x \in X$.

Therefore the score set of D is $S = \{a, ad, ad^2, \ldots, ad^p, ad^{p+1}\}$ which shows that the algorithm works for $n = p + 1$. Hence the algorithm is valid for all a, d and n satisfying the hypothesis of Theorem **??**.

The recursive procedure GEOMETRIC runs n times and during its i^{th} run the procedure adds $O(ad^{n+1-i})$ arcs to the oriented graph D. The overall complexity of the algorithm is therefore $O(nad^n)$.

As noted in Theorem **??**, there exists no 1-tournament when either $a = 1,\ d = 2,\ n > 0$ or $a = 1,\ d = 3,\ n > 0$. It is quite interesting to investigate these exceptional cases as it provides more insight into the problem.

Let us assume that $\mathbf{S} = \{1, 2, 2^2, \ldots, 2^n\}$ is a score set of some oriented graph D for $n > 0$. Then there exist positive integers, say $x_1, x_2, x_3, \ldots, x_{n+1}$ such that

$$S_1 = [1^{x_1}, 2^{x_2}, \ldots, (2^2)^{x_3}, \ldots, (2^n)^{x_{n+1}}$$

is the score sequence of D. Therefore, by relations (30.4) of score sequences of 1-tournaments, we have

$$x_1 + 2x_2 + 2^2x_3 + \cdots + 2^n x_{n+1} = \left(\sum_{i=1}^{n+1} x_i\right)\left(\sum_{i=1}^{n+1} x_i - 1\right),$$

which implies that x_1 is even. However, x_1 is a positive integer, therefore $x_1 \geq 2$. Let the scores be $a_1 = 1$, $a_2 = 1$ and $a_3 \geq 1$. By inequalities (30.4) $a_1 + a_2 + a_3 \geq 3(3-1) = 6$, or in other words, $a_3 \geq 4$. This implies that $x_2 = 0$, a contradiction.

The proof of the other exceptional case ($\mathcal{S} = \{1, 3, 3^2, \ldots, 3^n\}$) is left as an exercise (Exercise 30.2-1).

Let $I_p = a_1 > a_2 > \cdots > a_p$for $1 \leq p \leq n$. The next algorithm takes the set $I = \{a_1 < a_2 < \cdots < a_n\}$ consisting of n nonnegative integers as input and recursively constructs an oriented graph $D(V, A)$ with the score set $S = \{s_1, s_2, \ldots, s_n\}$ where the scores s_i are of the form 30.5.

ADDING-CONSTRUCTION2(n, I_n)

if $\quad n = 0$
$\quad\quad D = N_{a_1+1}$
$\quad\quad$ **return** D
else $\quad D^{(1)} =$ ADDING-CONSTRUCTION$(n-1, I_{n-1})$
$\quad\quad D = D^1 \oplus N_{a_{n+1}-a_n}$
$\quad\quad$ add arcs to D such that $N_{a_n - a_{n-1}}$ dominates $D^{(1)}$
return D

The running time of Adding-Construction2 is ???????

Example 30.3 ?????

Algorithm description

If $n = 1$, the algorithm returns the null digraph N_{a_1+1}. Each vertex of N_{a_1+1} has the score $a_1 + 1 - 1 + 0 - 0 = a_1 = a'_1$. Therefore the score set of N_{a_1+1} is $S = \{a'_1\}$ as required.

We prove the correctness of Adding-Construction2 in general by induction on n. Assume that the algorithm is valid for $n = 1, 2, \ldots, p$, for some integer $p \geq 2$. We show that the algorithm is also valid for $n = p + 1$. Let $a_1, a_2, \ldots, a_{p+1}$ be nonnegative integers with $a_1 < a_2 < \cdots < a_{p+1}$. Since $a_1 < a_2 < \cdots < a_p$, by the induction hypothesis, the algorithm returns an oriented graph $D^{(1)}$ on a_p+1 vertices with score set $\{a'_1, a'_2, \ldots, a'_p\}$, where a'_i is given by equations (30.5). That is, score set of $D^{(1)}$ is $\{a_1, a_1 + a_2 + 1, a_2 + a_3 + 1, \ldots, a_{p-1} + a_p + 1\}$. So a_1, $a_1 + a_2 + 1$, $a_2 + a_3 + 1, \ldots, a_{p-1} + a_p + 1$ are the distinct scores of the vertices of D. Let X be the vertex set of $D^{(1)}$ so that $|X| = a_p + 1$. Since $a_{p+1} > a_p$, $a_{p+1} - a_p > 0$, the algorithm constructs a new oriented graph $D = D^{(1)} \oplus N_{p+1}$ with vertex set $V = X \cup Y$, where Y is the vertex set of N_{p+1} and $|Y| = a_{p+1} - a_p$. Arcs are added to D such that there is an arc directed from each vertex in Y to every vertex in X. Thus D has $|V| = |X| + |Y| = a_p + 1 + a_{p+1} - a_p = a_{p+1} + 1$ vertices. The distinct score of vertices in X are $a_1 + |Y| - |Y| = a_1 = a'_1, a_1 + a_2 + 1 + |Y| - |Y| = a_1 + a_2 + 1 = a'_2, a_2 + a_3 + 1 + |Y| - |Y| = a_2 + a_3 + 1 = a'_3, \ldots, a_{p-1} + ap + 1 + |Y| - |Y| = a_{p-1} + a_p + 1 = a'_p$, while $a_y = |X| - 1 + |V| - 0 = a_{p+1} + 1 - 1 + a_p + 1 = a_p + a_{p+1} + 1 = a'_{p+1}$ for all $y \in Y$.

Therefore the score set of D is $S = \{s_1, s_2, \ldots, s_p, s_{p+1}\}$ which proves the validity of algorithm for $n = p + 1$. Hence by induction, Adding-Construction2 is valid for all n.

The analysis of computational complexity of General-Construction is left as an exercise (Exercise 30.2-3).

Exercises

30.2-1 Prove that there exists no oriented graph with score set $\{1, 3, 3^2, \ldots, 3^n\}$ for any $n > 0$.

30.2-2 Prove that if $n \geq 2$ and $\mathcal{S} = \{1, a_2, a_3, \ldots, a_n\}$, then $a_2 \geq 4$.

30.2-3 Adding-Construction is a recursive algorithm. Analyse its running time and compare its performance with the performance of Geometric-Construction.

30.2-4 Implement Adding-Construction in a suitable programming language and use it to construct an oriented graph with score set $\{2, 4, 8\}$. Write the score sequence of your oriented graph.

30.2-5 Implement Adding-Construction in a suitable programming language and use it to construct an oriented graph with score set $\{1, 4, 6, 9\}$. Write the score sequence of your oriented graph.

30.2-6 Give a proof of Lemma 30.6.

30.2-7 For any nonnegative integer n, what is the score set of the regular tournament T_{2n+1} when considered as an oriented graph.
30.2-8 Determine the score set of the oriented graph $D = T_3 \oplus T_5$, where T_5 dominates T_3, i.e., there is an arc directed from every vertex of T_5 to every vertex of T_3.
30.2-9 Write an $O(n)$ algorithm to determine the score set of directed cycles (i.e., cycles with directed edges). How can we make this algorithm work for directed wheels (note that a *wheel* is a cycle with an additional vertex joined to all the vertices on the cycle).

30.3. Unicity of score sets

***k*-tournaments** (multitournaments) are directed graphs in which each pair of vertices is connected with exactly k arcs.

Reid formulated the following conjecture in [38].

Conjecture 30.7 *Any set of nonnegative integers is the score set of some 1-tournament T.*

Using Landau's theorem this conjecture can be formulated in the following arithmetic form too.

Conjecture 30.8 *If $0 \le r_1 < r_2 < \cdots < r_m$, then there exist such positive integers $x_1, x_2, \ldots, x_m$, that*

$$\sum_{i=1}^{j} x_i r_i \ge \frac{(\sum_{i=1}^{j} x_i)(\sum_{i=1}^{j} x_i - 1)}{2}, \ j \in [1:m]$$

and

$$\sum_{i=1}^{m} x_i r_i = \frac{(\sum_{i=1}^{m} x_i)(\sum_{i=1}^{m} x_i - 1)}{2}.$$

In this case we say that the sequence $\mathbf{s} = \langle s_1, \ldots, s_n \rangle = \langle r_1^{x_1}, \ldots, r_m^{x_m} \rangle$ realizes the sequence $\mathbf{r} = \langle r_1, \ldots, r_m \rangle$ or $\mathbf{s}$ is a solution for $\mathbf{r}$.

Reid gave a constructive proof of his conjecture for sets containing one, two or three elements [38].

Later Hager published a constructive proof for sets with four and five elements [12] and Yao [46] published the outline of a nonconstructive proof of the general case.

A score set is called ***k*-unique,** if there exists exactly 1 score sequence of k-tournaments generating the given set. In the talk we investigate the following questions:

1. characterization of the unique score sets of 1-tournaments;
2. extension of the Reid's conjecture to 2-tournaments.

30.3.1. 1-unique score sets

At first we formulate a useful necessary condition.

Lemma 30.9 *(Iványi and Phong, 2004) If $k \geq 1$, then for any (n,k)-tournament holds that the sequence $\mathbf{s}$ is a solution for $\mathbf{r}$, then in the case $m = 1$ we have*

$$n = 2r_1 + 1 \tag{30.6}$$

and in the case $m \geq 2$ we have

$$\frac{2r_1}{k} + 1 < n < \frac{2r_m}{k} + 1 \tag{30.7}$$

and

$$n \geq r_m + 1\,. \tag{30.8}$$

Proof If

■

This lemma implies the exact answer for the case $m = 1$.

Corollary 30.10 *(Iványi and Phong, 2004) If $\mathbf{r} = \langle r_1 \rangle$, then exactly the sequence $\mathbf{s} = \langle r_1^{2r_1+1} \rangle$ is a solution for $\mathbf{r}$.*

Proof Lemma **??** implies that only this solution is acceptable. One can check that it satisfies the required inequality and equality. ■

Now we present a useful method of the investigation of the uniqueness. Let $\mathbf{r} = \langle a, a+d \rangle$. Then according to the Reid-equality we get

$$2ax + 2(a+d)y = n(n-1)$$

implying

$$y = \frac{n(n-2a-1)}{2d}\,. \tag{30.9}$$

But here only the values $n = 2a + 1 + i$ ($i \in [1, 2d-1]$) are permitted where

$$i \geq d + 1 - a\,. \tag{30.10}$$

By substitution $a = (q-1)d$ from (30.9) we get

$$y = \frac{(2qd - 2d + 2r + 1 + i)i}{2d}\,. \tag{30.11}$$

Here y must be an integer, so transform this formula into

$$y = i(q-d) + \frac{i(2r+1+i)}{2d}\,. \tag{30.12}$$

Theorem 30.11 *If $0 \le a < b$, then there exist positive integers x and y satisfying*

$$ax \ge \frac{x(x-1)}{2}$$

and

$$ax + by = \frac{(x+y)(x+y-1)}{2} \, .$$

In the following cases there is only one solution:

- *$a = 0$;*
- *$d = 1$;*
- *$d = 2$.*

In the following case there are at least two solutions:

- *d is odd and $3 \le d \le a$.*

Proof a) Existence of a solution. Let $d = b - a$ and $i = 2d - 2r - 1$. Then $n = 2(b - r)$, $y = q(2d - 2r - 1)$, $x = q(2r + 1)$ satisfy all requirements.

b) Uniqueness. If $a = 0$, then $d = b$, $q = 1$ and y is integer only if $i = 2b - 1$. So we get the unique $\langle 0^1, b^{2b-1} \rangle$ solution.

If $d = 1$, then only $i = 1$ is permitted, implying the unique solution $\langle a^b, b^b \rangle$.

If $d = 2$ or d is odd, then we also can analyse formula (30.12). ■

This theorem left open the case when the difference d is odd and the investigated set is sparse and also the case when the difference is an even number greater then 2.

30.3.2. 2-unique score sets

Now we present a new form of Reid-problem for 2-tournaments.

For a fixed sequence $\mathbf{q}[m] = \langle q_1, \ldots, q_m \rangle$ with $q_1 < \cdots < q_m$ of positive integers, we shall denote by $\mathcal{G}(\mathbf{q}[m])$ the set $\mathcal{G}$ of sequences $\mathbf{g} = \langle g_1, \ldots, g_m \rangle$ such that

$$\sum_{i=1}^{k} q_i g_i \ \ge \ \left(\sum_{i=1}^{k} g_i \right)^2 , \quad k \in [1 : m-1]$$

and

$$\sum_{i=1}^{m} q_i g_i \ = \ \left(\sum_{i=1}^{m} g_i \right)^2 .$$

Here we also say that $\mathbf{g}$ is a solution for $\mathbf{q}$.

We wish to give necessary and sufficient conditions for $\mathbf{q}[m]$ to have a solution that is a nonempty $\mathcal{G}(\mathbf{q}[m])$.)

Theorem 30.12 *For the sequence $\mathbf{q}[1] = \langle q_1 \rangle$, we have $\mathcal{G}(\mathbf{q}[1]) = \langle q_1 \rangle$.*

Proof If $\mathbf{q}[1] = \langle q_1 \rangle$, then it is obvious that the solution of $q_1 g_1 = g_1^2$ is given in the form $g_1 = q_1$. Hence we have $\mathcal{G}(\mathbf{q}[1]) = \langle q_1 \rangle$. ■

Theorem 30.13 *Let $\mathbf{q}[2] = \langle q_1, q_2 \rangle$ be a sequence of positive integers with $d = q_2 - q_1 > 0$. Then $\mathcal{G}(\mathbf{q}[2]) \neq \emptyset$ if and only if either $d \not| (q_1, q_2)$ or $d|(q_1, q_2)$ and there is a prime p such that $p^2|d$.*

Proof According to the definition of $\mathcal{G}(\mathbf{q}[m])$, we need only find positive integers g_1, g_2 such that $q_1 \geq g_1$ and $q_1 g_1 + q_2 g_2 = (g_1 + g_2)^2$.

Let $q,\ r$ be integers for which $q_2 = qd + r$, where $0 \leq r < d$. If $d \not| (q_1, q_2)$, then $r \neq 0$ and let $g_1 = rq$ and $g_2 = q_2 - r(q+1)$. Hence we have

$$
\begin{aligned}
g_1 = rq = r\frac{q_2 - r}{q_2 - q_1} &= r + r\frac{q_1 - r}{q_2 - q_1} \\
&< r + (R_1 - r) = R_1 \,, \\
&g_2 = R_2 - r(q+1) = \\
&q_2 - (q_2 - r)\frac{r}{q_2 - q_1} - r \\
&> q_2 - (q_2 - r) - r = 0
\end{aligned}
$$

and

$$
\begin{aligned}
q_1 g_1 + q_2 g_2 &= q_1 r q + q_2^2 - q_2 r(q+1) \\
&= q_2^2 + r(q_1 q - q_2 q + q_2) - 2q_2 r = \\
&= (q_2 - r)^2 = (g_1 + g_2)^2 .
\end{aligned}
$$

Now assume that $d|(q_1, q_2)$ and there is a prime p such that $p^2|d$. In this case $r = 0$ and we choose $g_1,\ g_2$ as follows:

$$
g_1 := \frac{q_2}{p} - \frac{d}{p^2} \quad \text{and} \quad g_2 := g_1(p-1) \,.
$$

It is obvious that

$$
g_1 > 0,\ \ g_2 > 0\ ,\ \ g_1 \leq R_1
$$

and

$$
\begin{aligned}
q_1 g_1 + q_2 g_2 &= g_1(q_1 + (p-1)q_2) \\
&= g_1(pq_2 - d) = \\
&= g_1 p^2 (\frac{q_2}{p} - \frac{d}{p^2}) = (g_1 p)^2 = (g_1 + g_2)^2 .
\end{aligned}
$$

Finally, assume that $d = 1$ or $d|(q_1, q_2)$ and d is the product of distinct primes. If there are positive integers g_1, g_2 such that $q_1 \geq g_1$ and $q_1 g_1 + R_2 g_2 = (g_1 + g_2)^2$, then we have $d|g_1 + g_2$ and

$$
\frac{1}{d}(g_1 + g_2)^2 - \frac{q_1}{d}(g_1 + g_2) = g_2 > 0 \,,
$$

$$
\frac{1}{d}(g_1 + g_2)^2 - \frac{R_2}{d}(g_1 + g_2) = -g_1 < 0 \,,
$$

consequently

$$\frac{q_2}{d} = \frac{q_1}{d} + 1 > \frac{g_1 + g_2}{d} > \frac{q_1}{d} \,.$$

This is impossible. ■

Theorem 30.14 Iványi, Phong, 2004 *Let* $\mathbf{q}[2] =< q_1, \ q_2 >$ *be the sequence of positive integers with conditions* $q_1 < R_2$, $(q_1, q_2) = 1$, $2q_1 > q_2$ *and* $d := q_2 - R_1$ *has* s *distinct prime factors. Then*

$$|\mathcal{G}(\mathbf{q}[2])| = 2^s - 1 \,.$$

Proof Since $d = q_2 - q_1 < q_1$ and $(q_1, q_2) = 1$, the congruence $x^2 \equiv q_2 x \pmod{d}$ has $2^s - 1$ solutions in positive integers less than d. For each solution x we set $g_1 = \frac{x(q_2 - x)}{d}$ and $g_2 = (d - x)\frac{q_2 - x}{d}$. One can check that g_1, g_2 satisfy conditions $q_1 \geq g_1$ and $q_1 g_1 + q_2 g_2 = (g_1 + g_2)^2$. ■

Exercises

30.3-1 How many ?

30.3-2 Design an algorithm

30.4. Kings and serfs in tournaments

Sociologists are often interested in determining the most *dominant* actors in a social network. Moreover, dominance in animal societies is an important theme in ecology and population biology. Social networks are generally modelled as digraphs with vertices representing actors and arcs representing dominance relations among actors. The concept of "king" is very closely related to dominance in digraphs. Kings and serfs were initially introduced to study dominance in round-robin competitions. These concepts were latter extended to more general families of digraphs such as multipartite tournaments, quasi-transitive digraphs, semicomplete multipartite digraphs and oriented graphs. In this section our focus will be on algorithmic aspects of kings and serfs in tournaments and their applications in majority preference voting.

A king in a tournament dominates every other vertex either directly or through another vertex. To make the idea more formal we define a *path* of length k from a vertex u to a vertex v in a tournament (or any digraph) as a sequence of arcs $e_1, e_2, \ldots, e_k$ where u is the initial vertex of e_1, v is the terminal vertex of e_k and the terminal vertex of e_i is the same as the initial vertex of e_{i+1}, for all $1 \leq i \leq k-1$. If there is a path of length 1 or 2 from a vertex u to a vertex v, then v is said to be ***reachable*** from u within two steps. Analogously, if there is a path of length $1, 2, \ldots$ or r from u to v then v is said to be reachable from u within r steps. Let T be an n-tournament. A vertex u in T is called an ***r-king,*** where $1 \leq r \leq n-1$, if every other vertex v in the tournament is reachable within r steps from u. A vertex u is called an ***r-serf*** if u is reachable within r if u is reachable within r steps from every

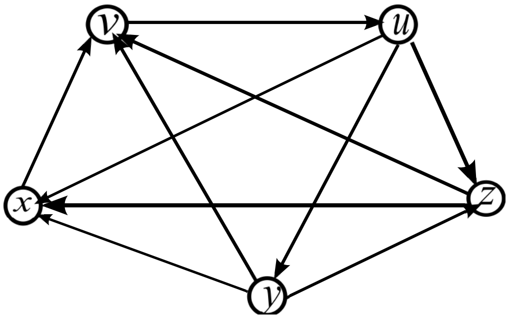

Figure 30.10 A tournament with three kings $\{u, v, y\}$ and three serfs $\{u, v, x\}$. Note that z is neither a king nor a serf and $\{u.v\}$ are both kings and serfs.

other vertex v in T. In particular, a 2-king is simply called a king and a 2-serf is called a serf.

S. B. Maurer introduced the dual terms of king and serf in a delightful exposition of a tournament model for dominance in flocks of chicken. In his influential series of papers on dominance in animal societies, H. G. Landau proved that every tournament has a king (although he did not use the word king). In fact, he showed the following.

Theorem 30.15 (Landau, 1953) *Every vertex of maximum score in a tournament is a king.*

The proof is quite intuitive. Suppose to the contrary that u is a vertex with maximum score in a tournament T and u is not a king. Then there exists another vertex v in T such that v is not reachable from u within 2 steps. But this means that u and all out-neighbours of u are reachable from v in 1 step and so $s(v) > s(u)$, a contradiction. Another classical result by J. W. Moon states that

Theorem 30.16 (Moon, 1968) *A tournament without transmitters (vertices with in-degree 0) contains at least three kings.*

It is natural to ask if the bound on the number of kings given in Theorem 30.16 is tight. The answer is yes, as demonstrated by the following example.

Example 30.4 Let T be a tournament with vertex set $\{v_1, v_2, \ldots, v_5\}$. Let us denote by (u, v), an arc directed from u to v. Suppose that the arc set of T consists of the arcs (v_3, v_5), (v_4, v_3), all arcs of the form (v_{j-1}, v_j), with $1 < j \le 5$ and all arcs of the form $(v_{j+2}, v_j), (v_{j+3}, v_j), \ldots, (v_n, v_j)$ with $j = 1, 2, 4$. Then it can be easily verified (Exercise 30.4-2) that T has no transmitters and v_2, v_3 and v_4 are the only kings in T.

K. B. Reid proved the existence of a tournament with an arbitrary number of vertices and an arbitrary number of kings, with few exceptions.

Theorem 30.17 (Reid, 1982) *For all integers $n \ge k \ge 1$ there exists a tournament on n vertices with exactly k kings except when $k = 2$ or when $n = k = 4$ (in which case no such n-tournament exists).*

Hence no tournament has exactly two kings. The above theorems can be stated just as well in terms of serfs. To see this, note that the *converse* T' of a tournament T, obtained by reversing the arcs of T, is also a tournament and that the kings and serfs of T and T' are interchanged.

The ***king set*** of a tournament consists of all kings in the tournament. We can define the ***serf set*** analogously. The problem of determining the king set of a tournament is very important both for theoretical and practical considerations. In voting theory literature, political scientists often refer to the uncovered set in majority preference voting. This uncovered set is actually the king set for the tournament whose vertices consist of the candidates to be elected and arcs represent the outcomes of the two-way race between candidates. Here we present a simple polynomial time algorithm for determining the king set of a tournament. Given an n-tournament T, let us define an $n \times n$ matrix D_T^+ as

$$(D_T^+)_{ij} = \begin{cases} 1 & \text{if } (v_i, v_j) \text{ is an arc of } T\,, \\ 0 & \text{otherwise}\,. \end{cases} \tag{30.13}$$

We call D_T^+, the *out-degree matrix* of T. When there is no danger of ambiguity we will drop the subscript T and simply denote the out-degree matrix by D^+. King-Set takes a tournament $T(V, A)$ as input, calculates the out-degree matrix D^+ of T and uses it to generate the king set K of T. Let O be the $n \times n$ zero matrix and let I be the $n \times n$ identity matrix.

King-Set(V, A)

$D^+ =$
$K = \emptyset$
for $i = 1$ **to** n
  for $j = 1$ **to** n
    if $(v_i, v_j) \in A$
      $(D^+)_{ij} = 1$
$M = I + D^+ + (D^+)^2$
$K = \{v_i \in V | \forall v_j \in V, (M)_{ij} \neq 0\}$
  N_n dominates $D^(1)$
return K

Algorithm description

The algorithm works on the same principle as the algorithm for finding the number of paths, from one vertex to another, in a digraph (Exercise 30.4-1 asks you to derive this algorithm). The (i, j) entry of the matrix $(D^+)^2$ is equal to the number of paths of length two from vertex v_i to vertex v_j (check this!). Therefore, the (i, j) entry of matrix $D^+ + (D^+)^2$ counts the number of paths of length one or two from v_i to v_j; and if vertex v_i is a king, all entries in the i^{th} row of $I + D^+ + (D^+)^2$ must be non-zero.

The computational complexity of Algorithm King-Set depends on the way $(D_T^+)^2$ is computed. If naive matrix multiplication is used, the algorithm runs in $\Theta(n^3)$ time. However, using the fast matrix multiplication by Coppersmith and

Winograd, the running time can be reduced to $O(n^{2.38})$. The Reader should note that by using the duality of kings and serfs, King-Set can be adapted for finding the serf set of a tournament.

King sets in majority preference voting

Kings frequently arise in political science literature. A ***majority preference voting*** procedure asks each voter to rank candidates in order of preference. The results can be modeled by a tournament where vertices represent the candidates and arcs point toward the loser of each two way race, where candidate u defeats candidate v if some majority of voters prefer u to v. Political scientists are often interested in determining uncovered vertices in the resulting tournament. A vertex u is said to *cover* another vertex v if u defeats v and also defeats every vertex that v defeats.

The covering relation is clearly transitive and has maximal elements, called ***uncovered vertices.*** An uncovered vertex u has the strategically important property that u defeats any other vertex v in no more than two steps, i.e., either

1. u defeats v or
2. there is some third alternative w such that u defeats w and w defeats v.

Thus an uncovered vertex is actually a king. In fact the uncovered set, consisting of all uncovered vertices, is precisely the set of all kings (see Exercise 30.4-8).

The idea behind finding kings in a tournament can be easily extended to finding r-kings for any positive integer r.

rKing-Set(V, A, r)

```
D⁺ = 0
K = ∅
for i = 1 to n
    for j = 1 to n
        if (v_i, v_j) ∈ A
            (D⁺)_ij = 1
M = I + D⁺ + ... + (D⁺)^r
K = {v_i ∈ V | ∀v_j ∈ V, (M)_ij ≠ 0}
return K
```

The above algorithm runs in $O(rn^3)$ if the matrix multiplications are performed naively, and in $O(rn^{2.38})$ time if fast matrix multiplication is incorporated.

As we have seen, kings dominate in tournaments. However, there exists a stronger notion of dominance in tournaments in the form of strong kings. Let us write $u \to v$ to denote that u defeats v in a tournament T, or in other words (u, v) is an arc of T. If U_1 and U_2 are disjoint subsets of vertices of T then we write $U_1 \to U_2$ to denote that all vertices in U_1 defeat all vertices in U_2. We define $B_T(u, v) = \{w \in V - \{u, v\} : u \to w \text{ and } w \to v\}$, where V denotes the vertex set of T. Let $b_T(u, v) = |B_T(u, v)|$. When no ambiguity arises, we drop the subscript T from the notation.

A vertex u in a tournament T is said to be a ***strong king*** if $u \to v$ or $b(u, v) > b(v, u)$ for every other vertex v of T.

Note that $b_T(u,v)$ is the number of paths of length two through which v is reachable from u. Therefore, $b_T(v_i,v_j) = ((D_T^+)^2)_{ij}$, where D_T^+ is the out-degree matrix of T.

Obviously, it is not true that every king is a strong king. For example, Figure 30.11 demonstrates a tournament with three kings, namely x, y and z. However, only x and y are strong kings as $b(z,x) < b(x,z)$. Figure 30.11 also shows that when searching for the most dominant vertex in real life applications, a king may not be the best choice (vertex z is a king, but it defeats only one vertex and is defeated by all other vertices). Therefore, choosing a strong king is a better option. This intuition is further confirmed by the fact that, in the probabilistic sense it can be shown that in almost all tournaments every vertex is a king.

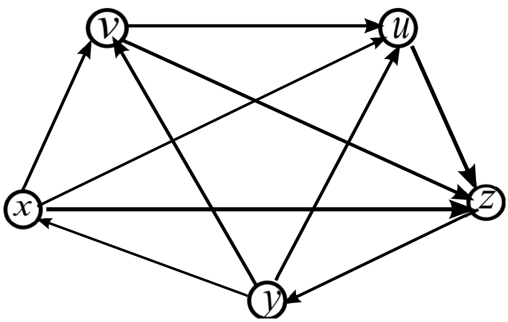

Figure 30.11 A tournament with three kings and two strong kings

We have already shown that every tournament has a king. We now prove that every tournament has a strong king.

Theorem 30.18 *(???, ????) Every vertex with maximum score in a tournament is a strong king.*

Proof Suppose u is a vertex with maximum score in a tournament T that is not a strong king. Then there is a vertex v in T such that $v \to u$ and $b(u,v) \le b(v,u)$. Let V be the vertex set of T. Define

$$W = \{w \in V - \{u,v\} : u \to w \text{ and } v \to w\} .$$

Then $s(u) = b(u,v) + |W|$ and $s(v) = b(v,u) + |W| + 1$. This implies that $s(u) < s(v)$, a contradiction. ■

The problem of finding strong kings is no harder than finding kings in tournaments. Like King-Set, we present a polynomial time algorithm for finding all strong kings in a tournament using the out-degree matrix D^+.

Strong-Kings(V, A)

```
D+ = 0
K = ∅
for i = 1 to n
    for j = 1 to n
        if (v_i, v_j) ∈ A
            D+_ij = 1
M = D+ + (D+)^2
K = {v_i ∈ V | ∀j(1 ≤ j ≤ n and j ≠ i), M_ij > M_ji}
return K
```

Strong-Kings has the same order of running time King-Set.

So far we have been focusing on finding certain type of dominant vertices (like kings and strong kings) in a tournament. Another very important problem is to construct tournaments with a certain number of dominant vertices. Maurer posed the problem of determining all 4-tuples (n, k, s, b) for which there exists a tournament on n vertices with exactly k kings and s serfs such that b of the kings are also serfs. Such a tournament is called an (n, k, s, b)-tournament. For example the tournament given in Figure **??** is a $(5, 3, 3, 2)$-tournament. Reid gave the following characterization of such 4-tuples.

Theorem 30.19 *Suppose that $n \geq k \geq s \geq b \geq 0$ and $n > 0$. There exists an (n, k, s, b)-tournament if and only if the following conditions hold.*

1. $n \geq k + s - b$,
2. $s \neq 2$ *and* $k \neq 2$,
3. *either* $n = k = s = b \neq 4$ *or* $n > k$ *and* $s > b$,
4. (n, k, s, b) *is none of* $(n, 4, 3, 2)$, $(5, 4, 1, 0)$, *or* $(7, 6, 3, 2)$.

However, the corresponding problem for strong kings has been considered only recently. For $1 \leq k \leq n$, a tournament on n vertices is called an (n, k)-tournament if it has exactly k strong kings. The construction of (n, k)- tournaments follows from the results proved by Chen, Chang, Cheng and Wang in 2004. The results imply the existence of (n, k)-tournaments for all $1 \leq k \leq n$ satisfying

$$k \neq n - 1, \text{ when } n \text{ is odd} \quad (30.14)$$
$$k \neq n, \qquad \text{when } n \text{ is even} . \quad (30.15)$$

Algorithm nk-Tournament takes positive integers n and k as input satisfying the constraints (26.2) and (26.3) and outputs an (n, k)-tournament and the set K of its strong kings. Also for any vertex u of a tournament T, we adopt the notation of Chen et al. in letting $O(u)$ (respectively, $I(u)$) denote the set of vertices reachable from u in one step (respectively, set of vertices from which u is reachable in one step). Note that $O(u)$ and $I(u)$ are often referred to as the first out-neighbourhood and first in-neighbourhood of u respectively.

nk-Tournament(n, k)

```
K = ∅
T = null digraph on n verices
if k is odd
   T = T_k
   K = {v_1, ..., v_k}
if n ≠ k
   for i = k + 1 to n
        V = V ∪ {v_i}
        A = A ∪ {(u, v_i) : u ∈ V − {v_i}}
if k is even
   T = T_{k−1}
   V = V ∪ {x, y, z}
   K = {v_1, ..., v_{k−3}, x}
   choose u ∈ V arbitrarily
   A = A ∪ {(v, x) : v ∈ O(u)}
   A = A ∪ {(x, v) : v ∈ {u, y} ∪ I(u)}
   A = A ∪ {(v, y) : v ∈ {u} ∪ I(u) ∪ O(u)}
   A = A ∪ {(v, z) : v ∈ {u} ∪ I(u)}
   A = A ∪ {(z, v) : v ∈ O(u)}
   if n ≠ k + 2
      for i = k + 1 to n
           V = V ∪ {v_i}
           A = A ∪ {(u, v_i) : u ∈ V − {v_i}}
return T, K
```

Algorithm description

The algorithm consists of performing two separate inductions to generate an (n, k)-tournament, one for odd k and one for even k.. If k is odd then we start by letting $T = T_k$, the regular tournament on k vertices (which always exists for odd k), and inductively add $n - k$ vertices to T that are defeated by all the vertices of T_k. Thus the resulting tournament has n vertices and k kings (the vertices of T_k). The construction for even k is a bit more involved. We start with $T = T_{k-1}$. Note that every vertex of T_{k-1} has score $m = \binom{n-4}{2}$. We then add three vertices x, y and z and several arcs to T_{k-1} such that for a fixed existing vertex u of T_{k-1}.

- $O(u) \to \{x\} \to \{u, y\} \cup I(u)$,
- $\{u\} \cup I(u) \cup O(u) \to \{y\} \to \{x, z\}$,
- $\{u\} \cup I(u) \to \{z\} \to O(u)$.

The resulting tournament T (illustrated in Figure 30.12) has $k + 2$ vertices with scores $s(x) = |I(x)| + 2 = m + 2$, $s(y) = 2$, $s(z) = |O(x)| = m$ and $s(v) = m + 2$,, for all vertices v of T_{k-1}. Now by Theorem 30.18 all vertices v of T_{k-1} and the new vertex x are strong kings of T, while y and z are not (Exercise 30.4-9). Thus T is a $(k + 2, k)$-tournament that can now be extended to an (n, k)-tournament by adding $n - k - 2$ more vertices that are defeated by all the existing vertices of T (just like

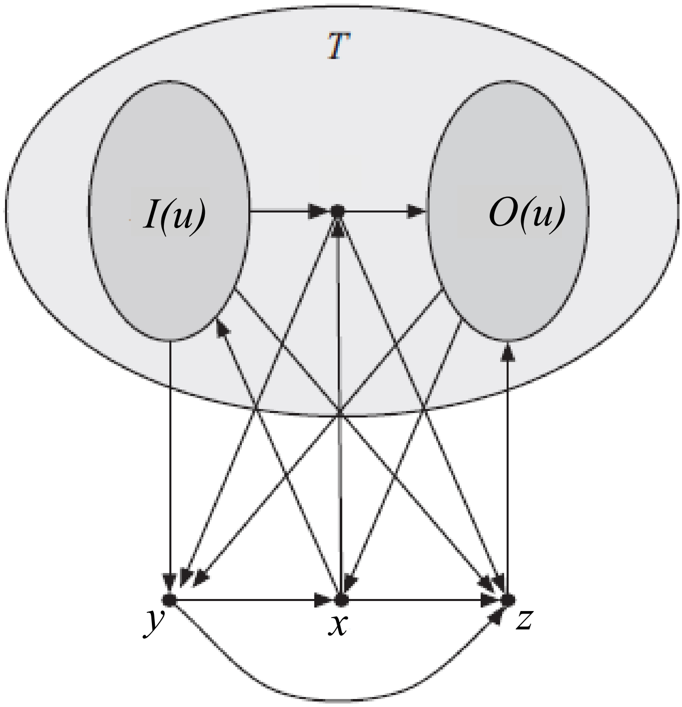

Figure 30.12 Construction of an (n, k)-tournament with even k.

in the case of odd k).

nk-TOURNAMENT runs in quadratic time as it takes $O(n^2)$ operations to construct a regular tournament and the remaining steps in the algorithm are completed in linear time.

Exercises

30.4-1 The out-degree matrix D^+ of an n-vertex oriented graph is an $n \times n$ matrix whose (i, j) entry is given by d_{ij} = number of arcs directed from v_i to v_j. Describe an algorithm based on the out-degree matrix for finding the number of paths of length $k < n$ between any two vertices of the graph.

30.4-2 Draw the tournament discussed in Example 30.4 and show that it has no transmitters and exactly three kings.

30.4-3 Using the 5-tournament in Example 30.4 give the construction of an n-tournament with no transmitters and exactly three kings.

30.4-4 For every odd number $n \geq 3$, give an example of an n-tournament, in which all vertices are serfs.

30.4-5 Prove that any tournament on 4 vertices contains a vertex which is not a king.

30.4-6 A *bipartite tournament* is an orientation of a complete bipartite graph. A vertex v of a bipartite tournament is called a 4-king[2] (or simply a king) if there is a directed path of length 4 from v to every other vertex of the tournament. Derive an algorithm to obtain all 4-kings in a bipartite tournament and compare its complexity with the complexity of r-Kings for finding r-kings in ordinary tournaments.

30.4-7 As the name suggests a multipartite tournament is an orientation of a complete multipartite graph. Extend the algorithm obtained in Exercise 30.4-6 to find all 4-kings in multipartite tournaments. Again compare the performance of your algorithms with r-Kings.

30.4-8 Prove that the uncovered set arising in majority preference voting is exactly the king set of the majority preference tournament.

30.4-9 Show that when k is even, the output of nk-Tournament has exactly k kings.

30.5. Weak kings in oriented graphs

In the previous section we studied dominance in tournaments and used the terms kings and strong kings to describe the dominant vertices in a tournament. However, in most practical applications the underlying digraph is not a tournament. Rather we are interested in determining dominant vertices in an oriented graph. For instance, in a social network, an arc (u, v) denotes that actor u has some relation with actor v.. Since most social relations (such as hierarchy relations) are irreflexive and asymmetric, a majority of social networks can be modelled as oriented graphs. Therefore, we would like to generalize the concept of dominance from tournaments to oriented graphs. In Section **??**, we have already defined kings and r-kings in the context of general digraphs. The same definitions are applicable to oriented graphs.

As stated in the beginning of the chapter, oriented graphs can be considered as round-robin competitions in which ties are allowed. Thus the the classical notion of king, that is a vertex that defeats every other vertex either directly or through another vertex, is too strong for oriented graphs. To overcome this difficulty, the study of the so-called "*weak kings*" was initiated in 2008 by S. Pirzada and N. A. Shah. Here we follow their notation. For any two vertices u and v in an oriented graph D,, one of the following possibilities exist.

1. An arc directed from u to v, denoted by $u(1-0)v$ (i.e., u defeats v).
2. An arc directed from v to u , denoted by $u(0-1)v$ (i.e., v defeats u).
3. There is no arc from u to v or from v to u , and is denoted by $u(0-0)v$ (i.e., there is a tie).

A ***triple*** in an oriented graph is an induced oriented subgraph with three vertices. For any three vertices u, v and w, the triples of the form $u(1-0)v(1-0)w(1-0)u$, $u(1-0)v(1-0)w(0-0)u$, $u(0-0)v(1-0)w(1-0)u$ or $u(1-0)v(0-0)w(1-0)u$ are said to be ***intransitive,*** while the triples of the form $u(1-0)v(1-0)w(0-1)u$,

[2]Several bipartite and multipartite tournaments have no 2-king or 3-king. However, a multipartite tournament with at least one vertex of in-degree zero contains a 4-king. Therefore it is logical to look for 4-kings in a multipartite tournament.

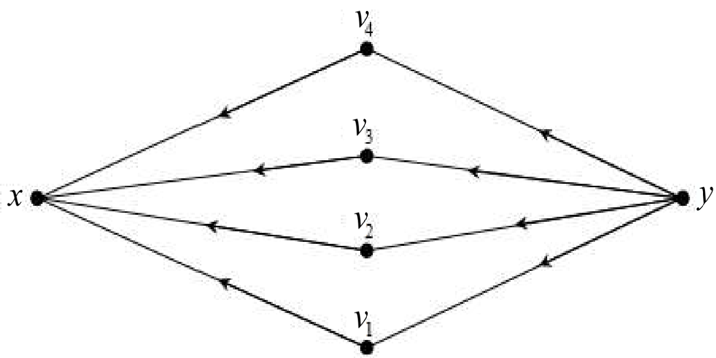

Figure 30.13 Six vertices and six weak kings.

$u(0-1)v(1-0)w(1-0)u$, $u(1-0)v(0-1)w(1-0)u$, $u(1-0)v(0-1)w(0-0)u$, $u(0-1)v(0-0)w(1-0)u$, $u(0-0)v(1-0)w(0-1)u$, $u(1-0)v(0-0)w(0-1)u$, $u(0-0)v(0-1)w(1-0)u$, $u(0-1)v(1-0)w(0-0)u$, $u(1-0)v(0-0)w(0-0)u$, $u(0-1)v(0-0)w(0-0)u$, $u(0-0)v(1-0)w(0-0)u$, $u(0-0)v(0-1)w(0-0)u$, $u(0-0)v(0-0)w(1-0)u$ or $u(0-0)v(0-0)w(0-1)u$ are said to be ***transitive.*** An oriented graph is said to be ***transitive*** if all its triples are transitive. The converse $\overline{D}$ of an oriented graph D is obtained by reversing each arc of D.

Let u and v be vertices in an oriented graph D such that either $u(1-0)v$ or $u(0-0)v$ or $u(1-0)w(1-0)v$ or $u(1-0)w(0-0)v$ or $u(0-0)w(1-0)v$ for some vertex w in D. Then v is said to be weakly reachable within two steps from u. If either $u(1-0)v$, or $u(1-0)w(1-0)v$ for some w in D, then v is reachable within two steps from u.

A vertex u in an oriented graph D is called a ***weak king*** if every other vertex v in D is weakly reachable within two steps from u. A vertex u is called a ***king*** if every other vertex v in D is reachable within two steps from u. A vertex u in an oriented graph D is called a ***weak serf*** if u is weakly reachable within two steps from every other vertex in D, and a vertex u in D is called a ***serf*** if u is reachable within two steps from every other vertex v in D.

We note that there exist oriented graphs on n vertices with exactly k kings for all integers $n \geq k \geq 1$, with the exception of $n = k = 4$. Theorem 30.17 guarantees the existence of complete oriented graphs (tournaments) with n vertices and exactly k kings for all integers $n \geq k \geq 1$, with the exceptions $k = 2$ and $n = k = 4$. An oriented graph D with exactly two kings can be constructed as follows. Let $V = \{v_1, v_2, \ldots, v_n\}$ be the vertex set of D, with arcs defined as $v_1(1-0)v_i$, for $i = 2, 4, \ldots, n$; $v_1(0-1)v_3$; $v_2(1-0)v_3$ and $v_2(1-0)v_i$, for $4 \leq i \leq n$; and for all other $i \neq j$, $v_i(0-0)v_j$. The vertices v_1 and v_3 are the only kings in D (Exercise 30.5-1).

There do not exist any complete or incomplete oriented graphs with 4 vertices and exactly 4 kings. Suppose to the contrary that this is the case and let D be

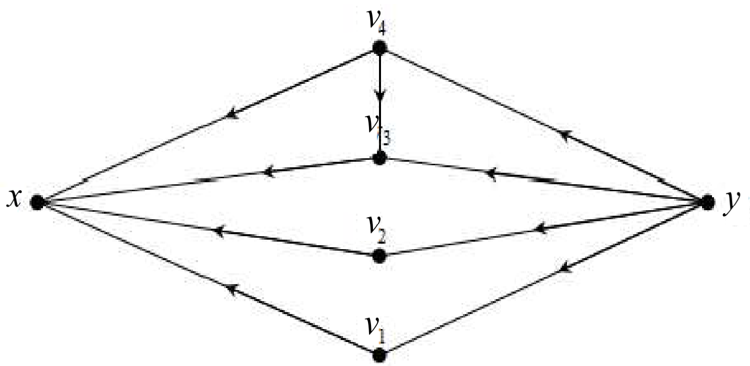

Figure 30.14 Six vertices and five weak kings.

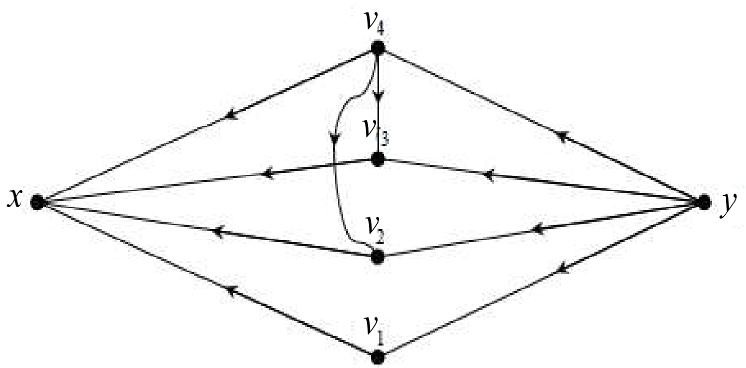

Figure 30.15 Six vertices and four weak kings.

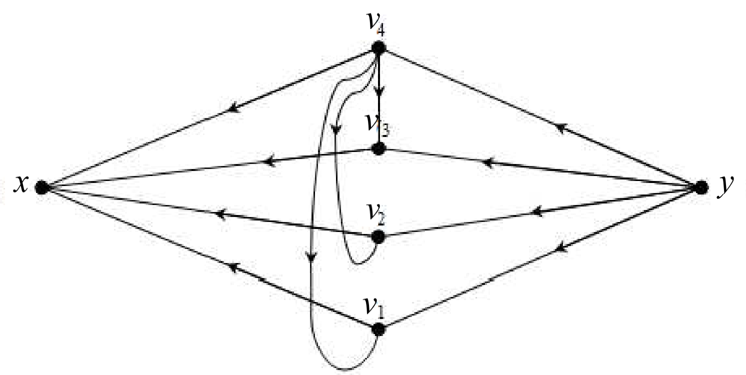

Figure 30.16 Six vertices and three weak kings.

the incomplete oriented graph with 4 vertices, all of whom are kings. Then D can be extended to a tournament on 4 vertices by inserting all the missing arcs with

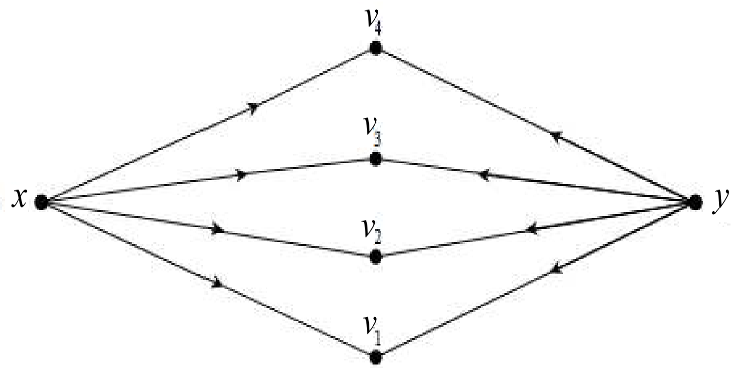

Figure 30.17 Six vertices and two weak kings.

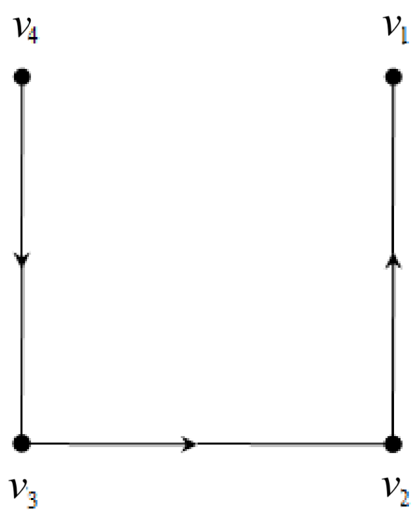

Figure 30.18 Vertex of maximum score is not a king.

arbitrary orientation. Clearly such a tournament contains 4 kings, which contradicts Theorem 30.17.

The rest of the section is aimed at investigating weak kings in oriented graphs as they present a more suitable notion of dominance in oriented graphs. The score of a vertex in an oriented graph was defined in Section **??**. Considering Theorem 30.15, it is natural to ask if a vertex of maximum score in an oriented graph is a king. The answer is negative as shown by the following example:

Example 30.5 Consider the oriented graph D shown in Figure 30.18. The scores of vertices v_1, v_2, v_3 and v_4 are respectively 2, 3, 3 and 4. Clearly, v_4 is a vertex of maximum score but is not a king as v_1 is not reachable within two steps from v_4. However, v_4 is a weak king.

Now consider the oriented graph D^* with vertices $u_1,$, u_2, u_3, u_4 and u_5, and arcs defined by $u_1(1-0)u_2$, $u_2(1-0)u_i$, for $i = 3,\ 4, q5$ and $u_i(0-0)u_j$ for all other $i \neq j$. Clearly, $s(u_1) = 5$, $s(u_2) = 6$, $s(u_3) = 3$, $s(u_4) = 3$, and $s(u_5) = 3$. Evidently, u_1 is a king in D^* whereas the vertex u_2 of maximum score is not a king.

However, we do have the following weaker result.

Theorem 30.20 *If u is a vertex with maximum score in a 2-tournament D, then u is a weak king.*

Proof Let u be a vertex of maximum score in D, and let X, Y and Z be respectively the set of vertices x, y, and z such that $u(1-0)x$, $u(0-0)y$, and $u(0-1)z$. Let $|X| = n_1$, $|Y| = n_2$ and $|Z| = n_3$. Clearly, $s(u) = 2n_1 + n_2$. If $n_3 = 0$, the result is trivial. So assume that $n_3 \neq 0$. We claim that each $z \in Z$ is weakly reachable within two steps from u. If not, let z_0 be a vertex in Z not weakly reachable within two steps from u. Then for each $x \in X$ and each $y \in Y$, $z_0(1-0)x$, and $z_0(1-0)y$ or $z_0(0-0)y$. In case $z_0(1-0)x$ and $z_0(1-0)y$ for each $x \in X$ and each $y \in Y$, then $s(z_0) \geq 2 + 2n_1 + 2n_2 = s(u) + n_2 + 2 > s(u)$. which contradicts the choice of u. If $z_0(1-0)x$ and $z_0(0-0)y$ for each $x \in X$ and each $y \in Y$, then $s(z_0) \geq 2+2n_1+n_2 = s(u)+2 > s(u)$, again contradicting the choice of u. This establishes the claim, and hence the proof is complete. ■

We now consider the problem of finding all weak kings in an oriented graph (as kings can be determined by applying Algorithm **??**). Let D^- and D^+ respectively denote the in-degree and out-degree matrix of an oriented graph $D(V, A)$ with n vertices. Also let O and J denote the $n \times n$ zero matrix and all-ones matrix respectively.

Weak-Kings(V, A)

$D^+ = 0$
$D^- = 0$
$K = \emptyset$
for $i = 1$ **to** n and $j = 1$ **to** n
  for $j = 1$ **to** n
    if $(v_i, v_j) \in A$
      $D^+_{ij} = 1$
    else if $(v_i, v_j) \in A$
      $D^-_{ij} = 1$
$M = J - D^-$
$M = D^+ + (D^+)^2$
$N = M + MD^+ + D^+M$
$K = \{v_i \in V \mid \forall v_j \in V, (N)_{ij} \neq 0\}$
return K

Algorithm **??** returns the set of all weak kings of an oriented graph. Exercise 30.5-3 asks you to prove that the algorithm works correctly and to determine its running time.

Indeed, it is also possible to extend Theorem 30.16 to weak kings in oriented graphs as an oriented graph D without transmitters (vertices of in-degree 0) has at least three weak kings. To see this let u be a vertex of maximum score in the oriented graph D. Clearly, by Theorem 30.20, u is a weak king. As D has no transmitters, there is at least one vertex v such that $v(1-0)u$. Let S be the set of these vertices v, and let v_1 be a vertex of maximum score in S. Let X, Y and Z respectively be the set of vertices x, y and z, other than u, with $v_1(1-0)x$, $v_1(0-0)y$ and $v_1(0-1)z$. Assume that $|X| = n_1$, $|Y| = n_2$, and $|Z| = n_3$ so that $s(v_1) = 2n_1 + n_2 + 2$. We note that all vertices of Z are weakly reachable within two steps from v_1. If this is not the case, let z_0 be a vertex which is not weakly reachable within two steps from v_1. Then $z_0(1-0)u$, and (a) $z_0(1-0)x$ and (b) $z_0(1-0)y$ or $z_0(0-0)y$ for each $x \in X$ and each $y \in Y$.

If for each x in X and each y in Y, $z_0(1-0)x$ and $z_0(1-0)y$, then $s(z_0) \geq 2n_1 + 2n_2 + 4 = s(v_1) + n_2 + 2 > s(v_1)$. This contradicts the choice of v_1. If for each x in X and each y in Y, $z_0(1-0)x$ and $z_0(0-0)y$, then $s(z_0) \geq 2n_1 + n_2 + 4 > s(v_1)$, again contradicting the choice of v_1. This establishes the claim, and thus v_1 is also a weak king.

Now let W be set of vertices w with $w(1-0)v_1$ and let w_1 be the vertex of maximum score in W. Then by the same argument as above, every other vertex in D is weakly reachable within two steps from w_1, and so w_1 is a weak king. Since D is asymmetric, and in D we have $w_1(1-0)v_1$ and $v_1(1-0)u$, therefore u, v_1 and w_1 are necessarily distinct vertices. Hence D contains at least three weak kings.

Although, no oriented graph with 4 vertices and exactly 4 kings exists, it is possible to generate an oriented graph on n vertices with exactly k weak kings, for all integers $n \geq k \geq 1$. The following algorithm constructs such an oriented graph.

k-Weak-Kings(n, k)

```
V = {x, y, u_1, u_2, ..., u_{n-2}}
x(0 - 0)y
if k > 2
   for i = 1 to n - 2
       u_i(1 - 0)x
       u_i(0 - 1)y

   for i = n - 3 downto k - 2
       u_{n-2}(1 - 0)u_i
   for i = k - 3 downto 1
       u_{n-2}(0 - 0)u_i
   K = {x, y, u_{n-2}} ∪ {u_i | i = 1, ..., k - 3}
   else if   k = 2
             for i = 1 to n - 2
                 x(1 - 0)u_i
                 y(1 - 0)u_i
             for j = 1 to n - 2
                 if i ≠ j
                    u_i(0 - 0)u_j
             K = {x, y}
        else x(1 - 0)u_i
             u_1(1 - 0)y
             for i = 2 to n - 2
                 u_1(1 - 0)u_i
                 x(1 - 0)u_i
                 y(1 - 0)u_i
             K = {u_1}
return V, A, K
```

Algorithm description

When $k = n$, the algorithm defines the arcs of a 2-tournament D with vertex set $V = \{x, y, u_1, u_2, \cdots, u_{n-2}\}$ as

$x(0-0)y$,

$u_i(1-0)x$ and $u_i(0-1)y$ for all $1 \le i \le n-2$,

$u_i(0-0)u_j$ for all $i \ne j$ and $1 \le i \le n-2$, $1 \le j \le n-2$,

Clearly, x is a weak king as $x(0-0)y$ and $x(0-0)y(1-0)u_i$ for all $1 \le i \le n-2$.. Also y is a weak king as $y(0-0)x$ and $y(1-0)u_i$ for all $1 \le i \le n-2$. Finally, every u_i is a weak king, since $u_i(0-0)u_j$, for all $i \ne j$ and $u_i(1-0)x$ and $u_i(1-0)x(0-0)y$. Thus D contains exactly n weak kings.

If $n = k-1$, the algorithm creates one additional arc $u_{n-2}(1-0)u_{n-3}$ in D. The resulting oriented graph contains exactly $n-1$ weak kings, since now u_{n-2} is not weakly reachable within two steps from u_{n-3} and so u_{n-3} is not a weak king.

If $n = k-2$ then the algorithm creates two additional arcs in D. namely $u_{n-2}(1-$

$0)u_{n-3}$ and $u_{n-2}(1-0)u_{n-4}$. Thus D now contains exactly $n-2$ weak kings, with u_{n-3} and u_{n-4} not being weak kings.

Continuing in this way, for any $3 \le k \le n$, the algorithm creates new arcs $u_{n-2}(1-0)u_i$ in D for all $k-2 \le i \le n-3$. The resulting graph D contains exactly k weak kings.

If $k = 2$, then D is constructed so that $x(0-0)y$, $x(1-0)u_i$. $y(1-0)u_i$ and $u_i(0-0)u_j$ for all $1 \le i \le n-2$, $1 \le j \le n-2$ and $i \ne j$. Thus D contains exactly two weak kings x and y.

Finally, D has exactly one weak king if it is constructed such that $x(0-0)y$, $u_1(1-0)x$, $u_1(1-0)y$ and $u_1(1-0)u_i$, $x(1-0)u_i$ and $y(1-0)u_i$ for all $2 \le i \le n-2$.

Due to the nested **for** loops the algorithm runs in $O(n^2)$ time.

Figure 30.13 shows a 6 vertex oriented graph with exactly 6 weak kings, Figure 30.14 shows a 6 vertex oriented graph with exactly 5 weak kings namely x, y, v_1, v_2 and v_4, Figure 30.15 shows a 6 vertex oriented graph with exactly 4 weak kings namely x, y. v_1 and v_4. Figure 30.16 shows a 6 vertex oriented graph with exactly 3 weak kings namely x, y and v_4 and Figure 30.17 shows a 6 vertex oriented graph with exactly 2 weak kings namely x and y.

The directional dual of a weak king is a weak serf, and thus a vertex u is a weak king of an oriented graph D if and only if u is a weak serf of $\bar{D}$, the converse of D. So by duality, there exists an oriented graph on n vertices with exactly s weak serfs for all integers $n \ge s \ge 1$. If $n = k \ge 1$, then every vertex in any such oriented graph is both a weak king and a weak serf. Also if $n > k \ge 1$, the oriented graph described in algorithm kWeakKings contains vertices which are both weak kings and weak serfs, and also contains vertices which are weak kings but not weak serfs and vice versa. These ideas give rise to the following problem. For what 4-tuples (n, k, s, b) does there exist an oriented graph with n vertices, exactly k weak kings, s weak serfs and that exactly b of the weak kings are also serfs? By analogy with (n, k, s, b)-tournaments, such oriented graphs are called (n, k, s, b)-oriented graphs. Without loss of generality, we assume that $k \ge s$. The following results by Pirzada and Shah address this problem.

Theorem 30.21 (Pirzada, Shah, 2008) *If $n > k \ge s \ge 0$, then there exists no (n, k, s, s)-oriented graph.*

Theorem 30.22 (Pirzada, Shah, 2008) *There exist (n, k, s, b)-oriented graphs, $n \ge k \ge s > b \ge 0$ and $n > 0$, $n \ge k + s - b$.*

Proof Let D_1 be the oriented graph with vertex set $\{x_1, y_1, u_1, u_2, \cdots, u_{k-b-2}\}$ and $x_1(0-0)y_1$, $u_i(1-0)x_1$, $u_i(0-1)y_1$ for all $1 \le i \le k-b-2$, and $u_i(0-0)u_j$ for all $i \ne j$.

Take the oriented graph D_2 with vertex set $\{x_2, y_2, v_1, v_2, \ldots, v_{b-2}\}$ and arcs defined as in D_1. Let D_3 be the oriented graph with vertex set $\{z_1, z_2, \ldots, z_{s-b}\}$ and $z_i(0-0)z_j$ for all i, j. Let D be the oriented graph $D_1 \cup D_2 \cup D_3$ (see Figure 30.19) with

$z_i(1-0)y_2$ for $1 \le i \le s-b$
$z_i(0-0)x_2$ for $1 \le i \le s-b$
$z_i(0-0)v_j$ for $1 \le i \le s-b$, $1 \le j \le b-2$

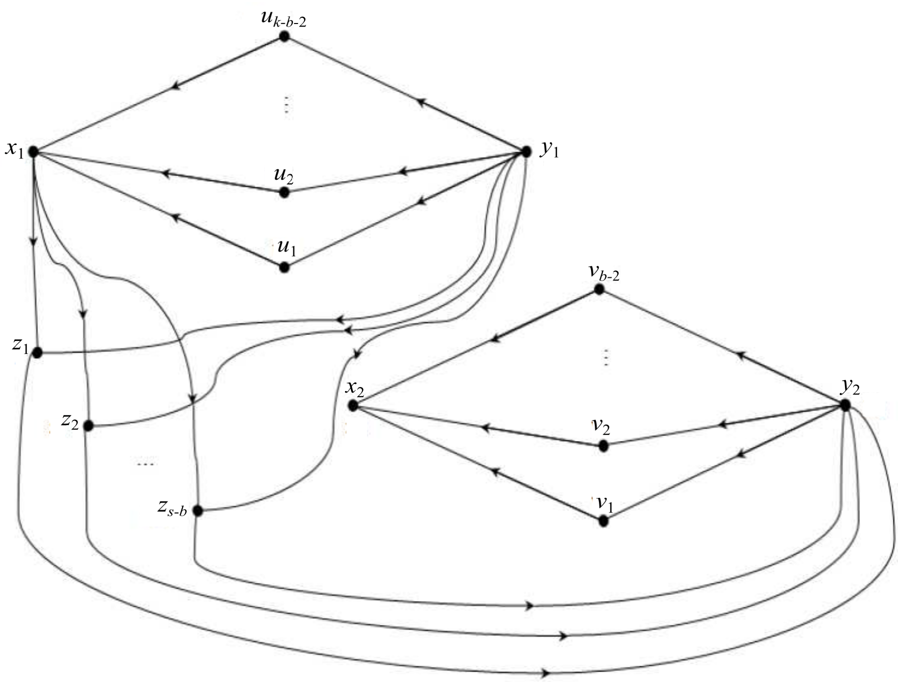

Figure 30.19 Construction of an (n, k, s, b)-oriented graph.

$x_1(1-0)z_i$, $y_1(1-0)z_i$ for $1 \le i \le s-b$
$u_r(0-0)z_i$ for $1 \le r \le k-b-2$, $1 \le i \le s-b$
$x_1(1-0)y_2$, $y_1(1-0)y_2$
$v_r(1-0)y_2$ for $1 \le r \le k-b-2$
$x_1(0-0)x_2$, $y_1(0-0)x_2$
$v_r(0-0)v_j$, for $1 \le r \le k-b-2$, $1 \le j \le b-2$.

Clearly D contains exactly k weak kings and the weak king set is $\{x_1, y_1\} \cup \{u_1, u_2, \ldots, u_{k-b-2}\} \cup \{x_2, y_2\} \cup \{v_1, v_2, \ldots, v_{b-2}\}$. D contains exactly s weak serfs with the weak serf set as $\{x_2, y_2\} \cup \{v_1, v_2, \ldots, v_{b-2}\} \cup \{z_1, z_2, \ldots, z_{s-b}\}$. Also from these k weak kings, exactly b are weak serfs. The weak king-serf set is $\{x_2, y_2\} \cup \{v_1, v_2, \ldots, v_{b-2}\}$. ■

Exercise 30.5-5 asks the reader to derive an algorithm for generating an (n, k, s, b)-oriented graph when the hypothesis of Theorem 30.22 is satisfied.

Exercises

30.5-1 Give an algorithm that generates an oriented graph with n vertices and exactly 2 kings. Prove the correctness of your algorithm.

30.5-2 Draw the graph D^* discussed in Example 30.5.

30.5-3 Prove that Weak-Kings2 is correct. Also determine its runtime.

30.5-4 Construct an oriented graph with six vertices and exactly one king.

30.5-5 Derive an algorithm that takes a 4-tuple (n, k, s, b) satisfying the hypothesis of Theorem 30.22 as input and generates an (n, k, s, b)-oriented graph. Analyze the performance of your algorithm.

Problems

30-1 Optimal reconstruction of score sets

In connection with the reconstruction of graphs the basic questions are the existence and the construction of at least one corresponding graph. These basic questions are often solvable in polynomial time. In given sense optimal reconstruction is usually a deeper problem.

a) Analyse Exercise 30.1-1 and try to find a smaller tournament with score set $\{0, 1, 3, 6, 10\}$.

b) Write a back-track program which constructs the smallest tournament whose score set is $\{0, 1, 3, 6, 10\}$.

c) Write a back-track program which constructs the smallest tournament arbitrary given score set.

d) Estimate the running time of your programmes.

Hint. Read Yoo's proof.

30-2 Losing set

We define the ***losing score*** of a vertex as the in-degree of the vertex. The *loosing score set* of a tournament is the set of in-degrees of its vertices.

a) Give an argument to show that any set of nonnegative integers is the loosing score set of some tournament.

b) Given a set $L = \{r_1, r_2, \ldots, r_n\}$ of nonnegative integers with $r_1 < r_2 - r_1 < r_3 - r_2 < \cdots < r_n - r_{n-1}$, write an algorithm to generate a tournament with loosing score set L.

30-3 Imbalance set

Let

30-4 Unicity

Let

30-5 generalized weak kings

Extend the definition of strong kings to (a, b)-tournaments and give sufficient conditions of the existence of the defined new types of kings.

30-6 generalized weak kings

Extend the definition of weak kings to (a, b)-tournaments and give sufficient conditions of the existence of the defined new types of kings.

Chapter Notes

Many classical ans several contemporary graph theory textbooks are available to Readers. Such books are e.g. the books of Dénes Kőnig [19], Claude Berge [4] and László Lovász [22]. However, there is a dearth of books focusing on recent advances in the theory of digraphs. The book due to Bang-Jensen and Gutin [2] probably comes closest and the Reader can refer to it for a comprehensive treatment of the theoretical and algorithmic aspects of digraphs.

The books by Harary, Norman and Cartwright [13], and Chartrand, Lesniak and Zhang [6, 7], Gross and Yellen [11] present introductory material on tournaments and score structures. Moon's book on tournaments [24] is also a good resource but is now out of print.

The books A. Schrijver [43] and A. Frank [10] contain reach material on optimization problems connected with directed graphs.

The algorithms discussed in this chapter are not commonly found in literature. In particular the algorithms presented here for constructing tournaments and oriented graphs with special properties are not available in textbooks. Most of these algorithms are based on fairly recent researchs on tournaments and oriented graphs.

Majority of the researches connected with score sequences and score sets were inspired by the work of H. G. Landau, K. B. Reid and J. W. Moon. For classical and recent results in this area we refer to the excellent surveys by Reid [38, 41, 42]. Landau's pioneering work on kings and score structure appeared in 1953 [20]. Reid stated his famous score set conjecture in [38]. Partial results were proved by M. Hager [12]. Yao's proof of Reid's conjecture appeared in English in 1989 [46]. The comment of Q. Li on Reid's conjecture and Yao's proof was published in 2006 [21]. The construction of a new special tournament with a prescribed score set is due to Pirzada and Naikoo [31]. The score structure for 1-tournaments was introduced by H. G. Landau [20] and extended for k-tournaments by J. W. Moon in 1963. This result of Moon later was reproved by Avery for $k = 2$ [1] and for arbitrary k by Kemnitz and Dolff [18]. Score sets of oriented graphs were investigated by Pirzada and Naikoo in 2008 [34].

Authors of a lot of papers investigated the score sets of different generalized tournament, among others Pirzada, Naikoo and Chisthi in 2006 (bipartite graphs), Pirzada and Naikoo in 2006 [32] (k-partite graphs), Pirzada and Naikoo in 2006 [33] (k-partite graphs). Pirzada, Naikoo and Dar analyzed signed degree sets of signed bipartite graphs [35].

The basic results on kings are due to K. Brooks Reid [39, 40, 41, 42] and Vojislav Petrović [5, 26, 27, 28, 29].

The problem of the unicity of score sequences was posed and studied by Antal Iványi and Bui Minh Phong [17]. Another unicity results connected with tournaments was published e.g. by P. Tetali, J. W. Moon and recently by Chen et al. [8, 9, 25, 45].

The term king in tournaments was first used by Maurer [23]. Strong kings were introduced by Ho and Chang [14] and studied later by Chen et al. [8, 9], while Pirzada and Shah [37] introduced weak kings in oriented graphs. The problems connected with 3-kings and 4-kings were discussed by Tan in [44] and the construction of tournaments with given number of strong kings by Chen et al. in [9].

The difference of the out-degree and of the in-degree of a given vertex is called *the imbalance* of the given vertex. The imbalance set of directed multigraphs were studied by Pirzada, Naikoo, Samee and Iványi in [36], while the imbalance sets of multipartite oriented graphs by Pirzada, Al-Assaf and Kayibi [30].

In connection with Problem 30-1 see [15, 16]

Problem 2
Problem 3
Problem 4
Problem 5
Problem 6

An interesting new direction is proposed by "L. B. Beasley, D. E. Brown, and. K. B. Reid in [3]: the problem is the reconstruction of tournaments on the base of the partially given out-degree matrix.

Bibliography

[1] P. Avery. Score sequences of oriented graphs. *Journal of Graph Theory*, 15:251–257, 1991. MR1111988 (92f:05042). 1540

[2] J. Bang-Jensen, G. Gutin. *Digraphs: Theory, Algorithms and Applications* (1st edition). Springer, 2009. MR2472389 (2009k:05001). 1540

[3] L. B. Beasley, D. E. Brown, K. B. Reid. Extending partial tournaments. *Mathematical and Computer Modelling*, 50(1–2):287–291, 2009. MR2542610 (2010i:05141). 1541

[4] C. Berge. *Graphs and Hypergraphs* (2nd revised edition). North-Holland, 1976. MR0384579 (52 #5453). 1539

[5] D. Brcanov, V. Petrović. Toppling kings in multipartite tournaments by introducing new kings. *Discrete Mathematics*, 310(19):2250–2554, 2010. 1540

[6] G. Chartrand, L. Lesniak. *Graphs and Digraphs* (4th edition). Chapman and Hall/CRC Press, 2005. MR2107429 (2005g:05001). 1540

[7] G. Chartrand, L. Lesniak, P. Zhang. *Graphs and Digraphs* (5th edition). Chapman and Hall/CRC Press, 2010. 1540

[8] A. Chen. *The strong kings of tournaments*. PhD thesis, National Taiwan University of Science and Technology, 2007. 1540

[9] A. Chen, J. Chang, Y. Cheng Y. Wang. The existence and uniqueness of strong kings in tournaments. *Discrete Mathematics*, 308(12):2629–2633, 2008. MR2410473 (2009c:05095). 1540

[10] A. Frank. *Connections in Combinatorial Optimization*. Oxford University Press, 2011 (to appear). 1540

[11] J. Gross, J. Yellen. *Handbook of Graph Theory* (2nd edition). CRC Press, 2006. MR2181153 (2006f:05001). 1540

[12] M. Hager. On score sets for tournaments. *Discrete Mathematics*, 58(1):25–34, 1986. 1518, 1540

[13] F. Harary, R. Norman, D. Cartwright. *An Introduction to the Theory of Directed Graphs*. John Wiley and Sons, Inc, 1965. 1540

[14] T. Ho, J. Chang. Sorting a sequence of strong kings in a tournament. *Information Processing Letters*, 87(6):317–320, 2003. 1540

[15] A. Iványi. Reconstruction of complete interval tournaments. *Acta Univ. Sapientiae, Informatica*, 1(1):71–88, 2009. 1540

[16] A. Iványi. Reconstruction of complete interval tournaments. II. *Acta Univ. Sapientiae, Mathematica*, 2(1):47–71, 2010. 1540

[17] A. Iványi, B. M. Phong. On the unicity of multitournaments. In *Fifth Conference on Mathematics and Computer Science* (Debrecen, June 9–12, 2004), 2004. http://compalg.inf.elte.hu/ tony/Publications. 1540

[18] A. Kemnitz, S. Dolff. Score sequences of multitournaments. *Congressus Numerantium*, 127:85–95, 1997. MR1604995 (98j:05072). 1540

[19] D. Kőnig. *Theorie der endlichen und unendlichen Graphen*. Akademische Verlagsgesellschaft, 1936. MR1550791. 1539

[20] H. G. Landau. On dominance relations and the structure of animal societies. III. the condition for a score sequence. *Bulletin of Mathematical Biophysics*, 15:143–148, 1953. MR0054933 (14,1000e). 1540

[21] Q. Li. Some results and problems in graph theory. *Annals of New York Academy of Sciences*, 576:336–343, 20. 1540

[22] L. Lovász. *Combinatorial Problems and Exercises*. Academic Press, 1979. MR0537284 (80m:05001). 1539

[23] S. B. Maurer. The king chicken theorems. *Mathematical Magazine*, 53:67–80, 1980. MR0567954 (81f:05089). 1540

[24] J. W. Moon. *Tournaments*. Holt, Rinehart, and Winston, 1963. MR0256919 (41 #1574). 1540

[25] J. W. Moon. The number of tournaments with a unique spanning cycle. *Journal of Graph Theory*, 53(3):303–308, 1982. MR0666798 (84g:05078). 1540

[26] V. Petrović. Bipartite kings. *Novi Sad Journal of Mathematics???*, 98(3):237–238, 1995. MR144406 (92h:05063). 1540

[27] V. Petrović. Kings in bipartite tournaments. *Discrete Mathematics*, 173(1–3):117–119, 1997. MR1468848 (98g:05067). 1540

[28] V. Petrović, C. Thomassen. Kings in k-partite tournaments. *Discrete Mathematics*, 98(3):237–238, 1991. MR144406 (92h:05063). 1540

[29] V. Petrović, M. Treml. 3-kings in 3-partite tournaments. *Discrete Mathematics*, 173(2–3):177–186, 1998. MR1468848 (98g:05067). 1540

[30] S. Pirzada, A. M. Al-Assaf, K. Kayibi, A. On imbalances in oriented multipartite graphs. *Acta Universitatis Sapientiae, Mathematica*, 3(1), 2011 (to appear). 1540

[31] S. Pirzada, T. A. Naikoo. On score sets in tournaments. *Vietnam Journal of Mathematics*, 34(2):157–161, 2006. 1540

[32] S. Pirzada, T. A. Naikoo. Score sets in oriented k-partite graphs. *AKCE International Journal of Graphs and Combinatorics*, 3(2):135–145, 2006. 1540

[33] S. Pirzada, T. A. Naikoo. Score sets in oriented k-partite tournaments. *Journal of Applied Mathematics and Computing*, 22(1–2):237–245, 2006. 1540

[34] S. Pirzada, T. A. Naikoo. Score sets in oriented graphs. *Applicable Analysis and Discrete Mathematics*, 2:107–113, 2008. 1540

[35] S. Pirzada, T. A. Naikoo, F. Dar. A note on signed degree sets in signed bipartite graphs. *Applicable Analysis and Discrete Mathematics*, 2(1):114–117, 2008. 1540

[36] S. Pirzada, T. A. Naikoo, U. Samee, A. Iványi. Imbalances in directed multigraphs. *Acta Universitatis Sapientiae, Mathematica*, 2(1):47–71, 2010. 1540

[37] S. Pirzada, N. A. Shah. Kings and serfs in oriented graphs. *Mathematica Slovaca*, 58(3):277–288, 2008. 1540

[38] K. B. Reid. Score sets for tournaments. *Congressus Numerantium*, 21:607–618, 1978. 1518, 1540

[39] K. B. Reid. Tournaments with prescribed numbers of kings and serfs. *Congressus Numerantium*, 29:809–826, 1980. MR0608478 (82d:05064). 1540

[40] K. B. Reid. Every vertex a king. *Discrete Mathematics*, 38(1):93–98, 1982. MR0676524 (84d:05091). 1540

[41] K. B. Reid. Tournaments: Scores, kings, generalisations and special topics. *Congressus Numerantium*, 115:171–211, 1996. MR1411241 (97f:05081). 1540

[42] K. B. Reid. Tournaments. In J. L. Gross, J. Yellen (Eds.), *Handbook of Graph Theory*, 156–184 pages. CRC Press, 2004. 1540

[43] A. Schrijver (Ed.). *Combinatorial Optimization. Vol A. Paths, Flows, Matchings; Vol. B. Matroids, Trees, Stable Sets. Vol. C. Disjoint Paths, Hypergraphs*. Springer, 2003. MR1956924 (2004b:90004a), MR1956925 (2004b:90004b), MR1956926 (2004b:90004c). 1540

[44] B. Tan. On the 3-kings and 4-kings in multipartite tournaments. *Discrete Mathematics*, 306(21):2702–2710, 2006. MR2263727 (2007f:05079). 1540

[45] P. Tetali. A characterization of unique tournaments. *Journal of Combinatorial Theory, Ser. B.*, 72(1):157–159, 1998. MR1604650 (98h:05086). 1540

[46] T. Yao. On Reid conjecture of score sets for tournaments. *Chinese Science Bulletin*, 10:804–808, 1989. MR1022031 (90g:05094). 1518, 1540

This bibliography is made by HBibTEX. First key of the sorting is the name of the authors (first author, second author etc.), second key is the year of publication, third key is the title of the document.

Underlying shows that the electronic version of the bibliography on the home page of the book contains a link to the corresponding address.

Subject Index

This index uses the following conventions. Numbers are alphabetised as if spelled out; for example, "2-3-4-tree" is indexed as if were "two-three-four-tree". When an entry refers to a place other than the main text, the page number is followed by a tag: *exe* for exercise, *exa* for example, *fig* for figure, *pr* for problem and *fn* for footnote.

The numbers of pages containing a definition are printed in *italic* font, e.g.

time complexity, *583*.

A
(a, b, n)-tournament, 1501
(a, b)-tournament, 1501
ADDING-CONSTRUCTION, *1516*

B
bipartite graph, 1540
bipartite tournament, 1529*exe*

C
comparison based ranking, 1501
cycle, 1518*exe*

D
directed cycle, 1518*exe*
directed graph, 1501
directed wheel, 1518*exe*

E
EVEN, *1507*

F
4-king, 1540

G
GEOMETRIC-CONSTRUCTION, *1514*

I
imbalance set, 1540
in-degree matrix, 1502
INSERTION-SORT, 1503
intransitive triple, *1530*

K
king, 1502, *1531*
kings and serfs in 1-tournaments, 1522–1530
king set, *1524*
KING-SET, *1524*
k-partite graph, 1540
k-partite tournaments, 1540
k-tournament, 1501, *1518*
k-unique score set, *1518*

L
losing score, 1539*pr*
losing score set, 1539*pr*

M
majority preference voting, *1525*
multipartite tournament, 1530*exe*
multitournament, 1518

N
nk-TOURNAMENT, *1528*
nk-WEAK-KINGS, *1536*

O
ODD, *1506*, 1508*exa*, 1509, 1510
1-tournament, 1501, 1511
oriented graph, *1511*
out-degree matrix, 1502

Q
QUICK-SET1, *1504*
QUICK-SET2, *1513*

R
reachable vertex, *1522*

r-king, *1522*
rKing-Set, *1525*
r-serf, 1522

S
Score-Reconstruction1, *1506*, 1509, 1511
score set, *1502*
score set of an oriented graph, *1513*
score sets in 1-tournaments, 1502–1511
score sets in oriented graphs, 1511–1518
serf, *1531*
serf set, *1524*
Set1, *1503*
signed bipartite graph, 1540
signed degree set, 1540
strong king, *1525*
Strong-Kings, *1527*

T
3-king, 1540
transitive oriented graph, *1531*
transitive triple, *1531*
triple, *1530*
intransitive triple, 1530
transitive triple, 1531
2-tournament, 1501, 1512

U
uncovered vertex, *1525*
unicity of score sets, 1518–1522

V
vertex cover, 1525

W
weak king, *1531*
Weak-Kings, *1534*
weak kings in 2-tournaments, 1530–1539
weak serf, *1531*
wheel, 1518*exe*

Z
$(0, 1)$-tournament, 1511

Name Index

This index uses the following conventions. If we know the full name of a cited person, then we print it. If the cited person is not living, and we know the correct data, then we print also the year of her/his birth and death.

A
Al-Assaf, Abdulaziz M., 1540
Avery, Peter, 1512, 1513

B
Bang-Jensen, Jørgen, 1540
Beasley, LeRoy B., 1541
Berge, Claude (1926–2002), 1539
Brown, David E., 1541

C
Cartwright, D., 1540
Chang, Jou-Ming, 1540
Chartrand, Gary, 1540
Chen, An-Hang, 1540
Chisthi, T. A., 1540
Coppersmith, Don, 1524
Cormen, Thomas H., 1502

D
Dar, F. A., 1540

F
Frank, András, 1540

G
Gutin, Gregory, 1540

H
Hager, Michael, 1504, 1540
Harary, Frank, 1540
Ho, Ting-Yem, 1540

I
Iványi, Antal Miklós, 1501, 1514, 1519, 1540

K
Kayibi, Koko K., 1540
Kőnig, Dénes (1884–1944), 1539

L
Landau, H. G., 1504, 1513, 1522, 1540
Leiserson, Charles E., 1502
Lesniak, Linda, 1540
Li, Qiao, 1505, 1540
Lovász, László, 1540

M
Maurer, Stephen B., 1522, 1540
Moon, John W., 1523, 1540

N
Naikoo, Tariq A., 1505, 1513, 1540
Norman, R. Z., 1540

P
Petrović, Vojislav, 1540
Phong, Bui Minh, 1514, 1519, 1540
Pirzada, Shariefuddin, 1505, 1513, 1530, 1537, 1540

R
Reid, K. Brooks, 1504, 1523, 1540, 1541
Rivest, Ronald L., 1502

S
Samee, U., 1540
Schrijver, Alexander, 1540
Shah, N. A., 1530, 1537, 1540
Stein, Clifford, 1502

T
Tetali, Prasad, 1540

W
Winograd, Shmuel, 1524

Y
Yao, Tianxing, 1504–1506, 1540

Z
Zhang, Ping, 1540