Hamiltonian paths in tournaments – A generalization of sorting DM19 notes fall 2006

Jørgen Bang-Jensen
Imada, SDU

30. august 2006

1 Introduction and motivation

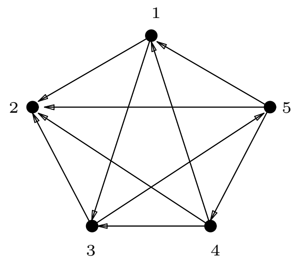
Tournaments which we will define mathematically in the next section are, besides being an interesting mathematical object, also interesting from a practical point of view. We shall show that the sorting problem is a special case of that of finding a hamiltonian path in a tournament. But let us start with a more amusing application of tournaments: Consider a soccer tournament in which every pair of distinct teams play each other exactly once and where ties are not allowed (that is in case of a tie after 90 minutes the game continues until one team has scored a goal, or, eventually the game is decided on penalties). The outcome of such a soccer tournament can be modeled as a mathematical tournament T by letting each team correspond to a vertex of T and letting T contain the arc $i{\rightarrow}j$ precisely when team i has beaten team j. The rule that there can be no ties ensures that T will be a mathematical tournament.

Since, as we shall prove below, every tournament has a hamiltonian path, it follows that one can always order the teams $1, 2, \dots, n$ in such a way that team i has beaten team $i + 1$ for $i = 1, \dots, n - 1$. Another interesting consequence is that there will always exist a team i such that for any other team j, either i has beaten j or i has beaten a team k which in turn has beaten j. You will be asked to prove this fact as one of the exercises.

The main purpose of this note is to show how various algorithms for sorting numbers can be converted into algorithms for finding hamiltonian paths in (mathematical) tournaments.

2 Some mathematics

A *tournament* is a directed graph $T = (V, A)$ such that for every pair of distinct vertices $x, y \in V$, precisely one of the arcs $x{\rightarrow}y$, $y{\rightarrow}x$ belongs to A.



	1	2	3	4	5
1	0	1	1	0	0
2	0	0	0	0	0
3	0	1	0	0	1
4	1	1	1	0	0
5	1	1	0	1	0

Figur 1: A tournament T on 5 vertices and the corresponding adjacency matrix $M(T)$. The numbering of the vertices of T corresponds to the numbering of the rows ands columns of $M(T)$.

Note that if $T = (V, A)$ is a tournament, then every subset $V' \subseteq V$ induces a *subtournament* of T and for every pair of distinct vertices $v, v' \in V'$, the arc $v \to v'$ is an arc in T' if and only if $v \to v'$ is an arc of T. We also call T' the *subtournament of T induced by V'*.

Given a tournament T on n vertices $v_1, \dots, v_n$ we can define an $n \times n$ matrix $M(T) = \{m_{ij}\}_{i,j=1,\dots,n}$, whose elements are all 0 or 1 and where $m_{ij} = 1 \iff m_{ji} = 0$ for $i \neq j$, and $m_{ii} = 0$, $i = 1, 2, \dots, n$. The matrix $M(T)$ is called the *adjacency matrix* of T and is defined by $m_{i,j} = 1 \iff v_i \to v_j$ is an arc of T.

Figure 1 shows a tournament on 5 vertices and the corresponding adjacency matrix.

A *path* in a digraph $D = (V, A)$ is a collection of vertices and arcs $P = x_1, x_1 \to x_2, x_2, x_2 \to x_3, \dots, x_{k-1} \to x_k, x_k$, such that the vertices $x_1, x_2, \dots, x_k$ are distinct. We say that P *starts* in x_1 and *ends* in x_k and call it an (x_1, x_k)-path. Similarly, a *cycle* is a collection of vertices and arcs $x_1, x_1 \to x_2, x_2, x_2 \to x_3, \dots, x_{k-1} \to x_k, x_k, x_k \to x_1$, such that the vertices $x_1, x_2, \dots, x_k$ are distinct. We will often just denote a path and a cycle as above by $x_1 \to x_2 \to \dots \to x_k$ respectively, $x_1 \to x_2 \to \dots \to x_k \to x_1$. The *length* of a path or a cycle is the number of arcs in it. A *hamiltonian path* in a digraph D is a path containing all vertices of D. For example, the path $3 \to 5 \to 4 \to 1 \to 2$ is a hamiltonian path in the tournament T in Figure 1.

Observe that if $x_1, x_2 \dots, x_n$ are distinct real numbers, then we can define a tournament TT_n by letting each x_i correspond to a vertex v_i and letting $v_i \to v_j$ be an arc of TT_n precisely when $x_i < x_j$. Then it is easy to see that TT_n does not contain any directed cycle and that TT_n will be the same for any choice of n distinct real numbers with the same relative order as $x_1, x_2 \dots, x_n$. We call TT_n the *transitive* tournament on n vertices. The name transitive comes from the fact that for all v_i, v_j, v_k such that $v_i \to v_j$ and $v_j \to v_k$ are arcs of TT_n, the arc $v_i \to v_k$ also belongs to TT_n. Furthermore, if we assume that $x_1 < x_2 < \dots < x_n$ then $v_1 \to v_2 \to \dots \to v_n$ is the unique hamiltonian path of T. As we shall show below, every tournament has a hamiltonian path. Thus it follows that the sorting problem is a special case of that of finding a hamiltonian path in a tournament. By this we mean the following: any algorithm for finding a hamiltonian path in a tournament can be used to sort a set of n numbers, since for tournaments defined from sets of distinct numbers as above there is precisely one hamiltonian path, the one corresponding to sorting the numbers in ascending order.

In Algorithm 2 below we shall make use of the following lemma.

Lemma 1 *Let $x_1 \rightarrow x_2 \rightarrow \ldots \rightarrow x_r$, $r \geq 2$, be a path in a tournament T and let x be a vertex not on this path. If both $x_1 \rightarrow x$ and $x \rightarrow x_r$ are arcs of T, then there exists an index $1 \leq i \leq r-1$, such that $x_i \rightarrow x$ and $x \rightarrow x_{i+1}$ are both arcs of T. Hence $x_1 \rightarrow \ldots \rightarrow x_i \rightarrow x \rightarrow x_{i+1} \rightarrow \ldots \rightarrow x_r$ is a path in T.*

Proof: Let i be the largest index such that $x_i \rightarrow x$ is an arc of T. Since $x \rightarrow x_r$ (implying that $x_r \rightarrow x$ is not an arc of T!) it follows that $i \leq r-1$ and by the maximality of i we must have $x \rightarrow x_{i+1}$. q.e.d.

3 Algorithms for finding a hamiltonian path in a tournament

Below we shall sketch two algorithms for finding a hamiltonian path in a tournament. These are based on two of the most fundamental principles that were covered in the course DM02, namely *divide and conquer* and *incremental algorithms.*

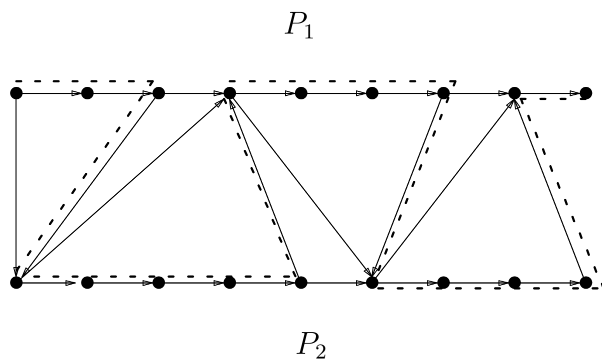
Algorithm 1:

This is an adaption of the well-know Merge-sort algorithm to tournaments:

1. Partition the vertex set of T into two disjoint sets V_1 and V_2 of equal or (if the number of vertices is odd) almost equal size . Denote by T_i the subtournament of T induced by V_i for $i = 1, 2$.

2. Find recursively a hamiltonian path P_i in T_i, $i = 1, 2$.

3. Merge the paths P_1 and P_2 into one path as follows: Let $P_1 = x_1 \rightarrow x_2 \rightarrow \ldots \rightarrow x_k$ and $P_2 = x_{k+1} \rightarrow x_{k+2} \rightarrow \ldots \rightarrow x_{k+l}$, $k = l$ or $k = l + 1$, be the two paths that we wish to merge into a hamiltonian path of T. Since T is a tournament, either $x_1 \rightarrow x_{k+1}$ or conversely. Assume without loss of generality below that $x_1 \rightarrow x_{k+1}$. Find an index i such that $x_i \rightarrow x_{k+1}$ and either $i = k$ or $x_{k+1} \rightarrow x_{i+1}$. If $i = k$ we are done, since then $x_k \rightarrow x_{k+1}$ and it is easy to construct a hamiltonian path of T. Suppose that $i < k$. Repeat the above process with the two paths $x_{i+1} \rightarrow \ldots \rightarrow x_k$ and $x_{k+1} \rightarrow \ldots \rightarrow x_{k+l}$. The merging is completed when one of the two remaining paths becomes empty. Figure 2 shows an example of the merging process.



Figur 2: An example of a merging of two paths. The dotted line indicates the resulting hamiltonian path.

Algorithm 2:

This is the equivalent of insertion sort for tournaments.

1. Let $v_1, \ldots, v_n$ be an arbitrary ordering of the vertices of the tournament T.

2. Consider the vertices v_1 and v_2. The unique arc between x_1 and x_2 forms a hamiltonian path in the subtournament induced by v_1 and v_2.

3. In the k'th step, $3 \leq k \leq n$, we proceed as follows: Let $x_1 \rightarrow \ldots \rightarrow x_{k-1}$ be the current hamiltonian path which the algorithm has found in the tournament induced by the vertices $v_1, \ldots, v_{k-1}$. Consider the arcs between the next vertex v_k and this path.

4. If $v_k \rightarrow x_1$, then $v_k \rightarrow x_1 \rightarrow \ldots \rightarrow x_{k-1}$ is a hamiltonian path in the subtournament induced by $v_1, \ldots, v_k$. Go to step 3 and consider the $(k+1)$'st vertex.

5. Otherwise, if $x_{k-1} \rightarrow v_k$ is an arc of T, then $x_1 \rightarrow \ldots \rightarrow x_{k-1} \rightarrow v_k$ is a hamiltonian path in the subtournament induced by $v_1, \ldots, v_k$. Go to step 3 and consider the $(k+1)$'st vertex.

6. (now we know that both $x_1 \rightarrow v_k$ and $v_k \rightarrow x_{k-1}$ are arcs of T, implying, by Lemma 1, that v_k can be inserted somewhere inside that path $x_1 \rightarrow \ldots \rightarrow x_{k-1}$). Find x_i, $1 \leq i \leq k-2$, such that $x_i \rightarrow v_k$ and $v_k \rightarrow x_{i+1}$ are both arcs. Let $x_1 \rightarrow \ldots \rightarrow x_i \rightarrow v_k \rightarrow x_{i+1} \rightarrow \ldots \rightarrow x_{k-1}$ be the new path and go to step 3 with the $(k+1)$'st vertex.

It is easy to see that the approach above can be converted into a proof by induction on the number of vertices of the following result.

Theorem 1 *Every tournament has a hamiltonian path.*

4 Exercises

1. Prove that every tournament $T = (V, A)$ has a vertex v such that v can reach every other vertex by a path of length at most 2 (i.e., for every $u \in V - v$, either $v \rightarrow u$ is an arc of T or there exists a vertex $z \in V - \{u, v\}$ such that $v \rightarrow z$ and $z \rightarrow u$ are both arcs of T. Hint: consider a vertex with the maximum number of arcs starting in it.

2. Let $T = (V, A)$ be a tournament and let P, Q be two paths, both of which start in x and end in y and which have no other common vertices (that is, they meet only at the ends).

   (a) Prove that T has a path R which starts in x, ends in y and which contains precisely those vertices which belong to one of the paths P, Q (that is $V(R) = V(P) \cup V(Q)$). Hint: prove that the paths can be merged as in Algorithm 1.

   (b) Describe a linear algorithm which, given two paths P, Q as above in a tournament, finds a path R as above. Hint: use the fact that one can construct R starting from the first vertex which is x and then proceeding forward along the two paths (as when one zips a fly), see Figure 2).

3. Describe how to implement Algorithm 1 and 2 such that the running times are $O(n \log n)$ and $O(n^2)$, respectively, when we measure the complexity as the number of times we need to look up the direction of an arc in the adjacency matrix.

4. A digraph $D = (V, A)$ is *strongly connected* if it contains and (x, y)-path for every choice of vertices $x, y \in V$. Prove that every strongly connected tournament has a hamiltonian cycle, that is, a cycle which contains all vertices of T. Hint: Start with a cycle C and conclude that if C is not a hamiltonian cycle, then, since T is strongly connected, one can extend the cycle C by adding one or more vertices. Here you can use Lemma 1.

5. Convert you proof above to an algorithm for finding a hamiltonian cycle in a tournament and give the complexity of your algorithm.