DIMACS Technical Report 2000-12
April 2000

An Efficient Sorting Algorithm for a Sequence of Kings in a Tournament

by

Jie Wu

Department of Computer Science and Engineering

Florida Atlantic University

Boca Raton, FL 33431

Li Sheng

Department of Mathematics and Computer Science

Drexel University

Philadelphia, PA 19104

DIMACS is a partnership of Rutgers University, Princeton University, AT&T Labs-Research, Bell Labs, NEC Research Institute and Telcordia Technologies (formerly Bellcore).

DIMACS is an NSF Science and Technology Center, funded under contract STC–91–19999; and also receives support from the New Jersey Commission on Science and Technology.

ABSTRACT

A king u in a tournament is a player who beats ($\rightarrow$) any other player v *directly or indirectly*. That is, either $u \rightarrow v$ or there exists a third player w such that $u \rightarrow w$ and $w \rightarrow v$. A sorting sequence of kings in a tournament of n players is a sequence of players, $S = (u_1, u_2, ..., u_n)$, such that $u_i \rightarrow u_{i+1}$ and u_i in a king in the sub-tournament T_{u_i} induced by $u_i, u_{i+1}, ..., u_n$ for $i = 1, 2, ..., n-1$. The existence of a sorting sequence of kings in any tournament is shown [3] where a sorting algorithm with a complexity of $\Theta(n^3)$ is given. In this paper, we present a constructive proof for the existence of a sorting sequence of kings of a tournament and propose an efficient algorithm with a complexity of $\Theta(n^2)$.

Keywords: King, sorting algorithm, tournament, median order, local median order.

A directed graph with a complete underlying graph is called a *tournament* [1], representing a tournament of n (≥ 1) players where every two players compete to decide the winner (and the loser) between them. A *king* u in a tournament is a player who beats ($\rightarrow$) any other player v *directly or indirectly*. That is, either $u \rightarrow v$ or there exists a third player w such that $u \rightarrow w$ and $w \rightarrow v$. A *sorting sequence of kings* [3] in a tournament of n players is a sequence of players, $S = (u_1, u_2, ..., u_n)$, such that $u_i \rightarrow u_{i+1}$ and u_i in a king in the sub-tournament T_{u_i} induced by $u_i, u_{i+1}, ..., u_n$ for $i = 1, 2, ..., n-1$. The existence of a sorting sequence of kings in any tournament is shown [3] where a sorting algorithm with a complexity of $\Theta(n^3)$ is given. In this paper, we present a constructive proof for the existence of a sorting sequence of kings of a tournament and propose an efficient algorithm with a complexity of $\Theta(n^2)$.

Lemma 1: ([2]) *Every tournament has a king.*

Lemma 2: *If u is a king for some tournament T and let $S \subseteq in(u) = \{v \in T : v \rightarrow u\}$, then u is still a king in the sub-tournament induced by $T - S$.*

Proof: We only need to consider the vertex $v \in T - S$ such that u beats v indirectly in T, i.e., $u \rightarrow w$ and $w \rightarrow v$. Clearly, $w \notin S$. Therefore, u still beats v indirectly in $T - S$. ■

Theorem 1: *Sorting sequence of kings exists in any tournament T of n players.*

Proof: We prove the theorem by induction on n. Clearly, it is true for $n = 1$. Assume that the statement is true for $n - 1$, we will show for the case of n. By Lemma 1 we can pick a king of T, say u, and by induction hypothesis, we can also assume that $S = (u_1, u_2, \cdots, u_{n-1})$ is a sorting sequence of kings of sub-tournament $T - \{u\}$. We shall show that u can be inserted into sequence S without changing any relative position of the vertices in S.

Suppose p ($1 \leq p \leq n-1$) is the first index such that $u \rightarrow u_p$ (such u_p always exists because u is a king of T). We shall show that $S' = (u_1, u_2, \cdots, u_{p-1}, u, u_p, u_{p+1}, \cdots, u_{n-1})$ is the sorting sequence of kings in T. Let $T_v(S')$ be the sub-tournament of T induced by v and all the vertices in S' that follow v. We need to show that

$$v \text{ is a king in } T_v(S') \text{ for all } v \in \{u_1, u_2, \cdots, u_{p-1}, u, u_p, u_{p+1}, \cdots, u_{n-1}\} \qquad (1)$$

Clearly, condition (1) is true for all $v \in \{u_p, u_{p+1}, \cdots, u_{n-1}\}$. By Lemma 2, condition (1) is also true for $v = u$. Now, we consider $v = u_i \in \{u_1, u_2, ..., u_{p-1}\}$. By induction hypothesis, u_i is a king of the sub-tournament induced by $T_{u_i}(S) = \{u_i, \cdots, u_{p-1}, u_p, \cdots, u_{n-1}\}$, together with $u_i \rightarrow u$, u_i is still a king of the sub-tournament induced by $T_{u_i}(S) \cup \{u\} = T_{u_i}(S')$.
■

Based on Theorem 1, we can easily derive an algorithm that successively inserts a vertex to a partial sorting sequence of kings. The key is to find a king in each sub-tournament. The following theorem provides an efficient way to determine such a king.

Theorem 2: *Let u be a vertex with the maximum out-degree in a tournament $T = (V, A)$. Then u is a king.*

Proof: Suppose u is not a king. Then there is a vertex v such that $(v, u) \in A$ and that $(v, w) \in A$ for every vertex $w \in out(u) = \{v \in T, u \rightarrow v\}$. This implies that $|out(v)| > |out(u)|$, a contradiction. ∎

We follow closely the proofs of Theorem 2 and Theorem 1 to generate a king sequence and a sorting sequence of kings in a tournament, respectively. The algorithm consists of three modules applied in sequence: OUT-DEGREE, KING-SEQUENCE, and KING-SORT. OUT-DEGREE computes the out degree of each vertex u and stores it in $O(u)$. KING-SEQUENCE generates a king sequence stored in an array B such that $B[i]$ is a king of sub-tournament $\{B[i], B[i+1], ..., B[n]\}$ for $i = 1, 2, ..., n$. KING-SORT successively inserts $B[i]$ into a sorting sub-sequence of kings $(B[i+1], B[i+2], ..., B[n])$ for $i = n-1, n-2, ..., 1$. Assume that $T = (V, A)$ is a given tournament such that $|V| = n$.

```
OUT-DEGREE
O(u) <-- 0, for each u ∈ V
for each e = (u, v) ∈ A
     do O(u) <-- O(u) + 1
```

```
KING-SEQUENCE
for i = 1 to n
     do B[i] <-- king, where O(king) = max_{v∈V}{O(v)}
        O(king) <-- -1
        for each e = (u, king) ∈ A
            do O(u) <-- O(u) - 1
```

```
KING-SORT
for i = n - 1 downto 1
     do for j = i to n - 1
            do if B[j + 1] -> B[j]
                 then exchange B[j] <--> B[j + 1]
                 else  return
```

Theorem 3: *The overall complexity of the algorithm is* $\Theta(|V|^2)$.

Proof: The complexity of OUT-DEGREE is $\Theta(|A|)$. In KING-SEQUENCE, the cost of decrementing $O(u)$ is $\Theta(|A|)$. The cost of searching for new kings in $|V|$ sub-tournaments is $\Theta(|V|^2)$. Note that at each round only one king is selected although several kings may exist. The complexity of KING-SORT is $\Theta(|V|^2)$. Therefore, the overall complexity is $\Theta(|V|^2 + |A|) = \Theta(|V|^2)$. ∎

Consider a sample tournament of six players $\{u_1, u_2, u_3, u_4, u_5, u_6\}$. Figure 1 shows the graph representation of the tournament. Applying the OUT-DEGREE algorithm, we have $(O(u_1), O(u_2), O(u_3), O(u_1), O(u_2), O(u_3)) = (4, 1, 4, 3, 2, 1)$. A step by step application of KING-SEQUENCE to generate $B[1...6]$ is shown as follows:

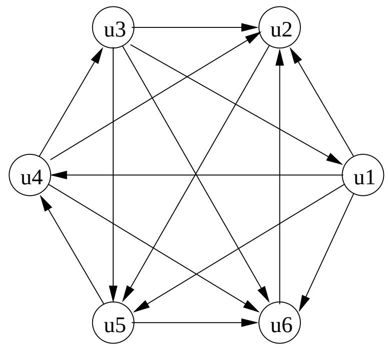

Figure 1: A sample example

$$\begin{array}{llllll} (\mathbf{4},1,4,3,2,1) & \xrightarrow{u_1} & (-1,1,\mathbf{3},3,2,1) & \xrightarrow{u_3} & (-1,1,-1,\mathbf{2},2,1) & \xrightarrow{u_4} \\ (-1,\mathbf{1},-1,-1,1,1) & \xrightarrow{u_2} & (-1,-1,-1,-1,\mathbf{1},0) & \xrightarrow{u_5} & (-1,-1,-1,-1,-1,\mathbf{0}) & \xrightarrow{u_6} \end{array}$$

Therefore, the resultant king sequence is $B[1...6] = [u_1, u_3, u_4, u_2, u_5, u_6]$. A step by step application of KING-SORT to $B[1...6]$ is shown as follows:

1. $u_1, u_3, u_4, u_2, u_5 \rightarrow u_6$
2. $u_1, u_3, u_4, u_2 \rightarrow u_5 \rightarrow u_6$
3. $u_1, u_3, u_4 \rightarrow u_2 \rightarrow u_5 \rightarrow u_6$
4. $u_1, u_4 \rightarrow u_3 \rightarrow u_2 \rightarrow u_5 \rightarrow u_6$
5. $u_1 \rightarrow u_4 \rightarrow u_3 \rightarrow u_2 \rightarrow u_5 \rightarrow u_6$

The final sorting sequence of kings is $u_1 \rightarrow u_4 \rightarrow u_3 \rightarrow u_2 \rightarrow u_5 \rightarrow u_6$. Note that in general the sorting sequence of kings is not unique. For example, $u_3 \rightarrow u_1 \rightarrow u_4 \rightarrow u_2 \rightarrow u_5 \rightarrow u_6$ is another sorting sequence of kings for Figure 1.

References

[1] J. A. Bondy and U. S. R. Murthy. *Graph Theory and Applications.* The Macmillan Press. 1976.

[2] H. G. Landau. On dominance relations and the structure of animal societies, III: The condition for score structure. *Bull. Math. Biophys.* **15**, 1953, 143-148.

[3] W. Lou, J. Wu, and L. Sheng. On the existence of a sorting sequence of kings in a tournament. *Thirty-First Southeastern International Conference on Combinatorics, Graph Theorey, & Computing.* March 2000, (Abstract only).