CSE 551 Homework 2

Due Date: 5th October (hard-copy in class submission, soft-copy blackboard submission)

Q1. The tournament method performs comparisons in the same way that tournaments are played. A tournament can be described as a tree diagram, where every leaf node contains one player, and at each subsequent level, the parent (internal node) of each pair contains the winner. Write an algorithm to find the second-best player in a list of *n* players in ($N + \log_2 N - 2$) comparisons (best player corresponds to the largest key in the list).

Q2. Let T be an array of *n* one-dimensional elements. Interchange the first *k* and the last *n-k* elements, without making use of an auxiliary array.
Solution: O(n).

Q3. You are to organize a tournament involving *n* competitors. Each competitor must play exactly once against each of his opponents. Moreover, each competitor must play exactly one match every day, with the possible exception of a single day when he does not play at all.

Consider the following scenarios:

1. If *n* is a power of 2, give an algorithm to construct a timetable allowing the tournament to be finished in *(n - 1)* days. Solution: O(n). You can use divide and conquer to get the optimal result. *
2. For any integer $n > 1$, give an algorithm to construct a timetable allowing the tournament to be finished in $n - 1$ days if *n* is even, or in *n* days if *n* is odd. Solution O(n). You can assume a list of players is given. Schedule tournaments in a circular fashion, in one loop, to get overall complexity as O(n).

*. The recurrence relation formed in Q3.1 is $T(n) = T(n/2) + n/2$. $T(2) = 1$. The base condition specifies that if there are 2 players, we can schedule the game in 1 day. $T(n/2)$ keeps on dividing the set of players and $n/2$ is the number of days taken to schedule the tournament when you recombine all players.