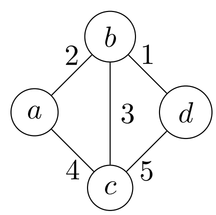
CSE421: Design and Analysis of Algorithms	April 22, 2021
Homework 4	
Shayan Oveis Gharan	Due: April 29th, 2021 at 23:59 PM

P1) (20 points) We have n individuals in a party some pairs are friends. Suppose friendship is a two-way relation, i.e., if i is a friend of j then j is a friend of i. Given an integer k design a polynomial time algorithm that outputs the largest subset S of these individuals such that each person in S has at least k friends in S. For example suppose $n = 4$ and we have the following friendships: $1 \leftrightarrow 2, 1 \leftrightarrow 3, 2 \leftrightarrow 3, 1 \leftrightarrow 4$ and $k = 2$. Then you should output $S = \{1, 2, 3\}$.

P2) Given a connected undirected weighted graph $G = (V, E)$ with positive weights on the edges, i.e., $c_e > 0$ for all e. Design a polynomial time algorithm to find the *largest weight* set of edges $F \subseteq E$ such that if we delete all edges of F the remaining graph is still connected. For simplicity assume that for any two edges $e, f \in E$, $c_e \neq c_f$. For example in the following example the optimum set F is $F = \{(a, c), (c, d)\}$.



P3) (10 points) Suppose you are choosing between the following three algorithms:

a) Algorithm A solves the problem by dividing it into seven subproblems of half the size, recursively solves each subproblem, and then combines the solution in linear time.

b) Algorithm B solves the problem by dividing it into twenty five subproblems of one fifth the size, recursively solves each subproblem, and then combines the solutions in quadratic time.

c) Algorithm C solves problems of size n by recursively solving four subproblems of size $n-4$, and then combines the solution in constant time.

In all cases you can assume it takes $O(1)$ time to solve instances of size 1. What are the running times of each of these algorithms? To receive full credit, it is enough to write down the running time.

P4) (20 points) Given a sequence of n numbers $a_1, \ldots, a_n$, we say this sequence is special if there is an integer $1 \leq m \leq n$ such that

- For all $1 \leq i < m$, $a_i < a_{i+1}$, and
- For all $m \leq i < n$, $a_i > a_{i+1}$.

For example, $1, 4, 3, 2$ is special. Given a special sequence of n numbers stored in an array $A[i] = a_i$, design an $O(\log n)$ time algorithm that outputs the largest number in this array, i.e., $A[m] = a_m$.

P5) **Extra Credit** The spanning tree game is a 2-player game. Each player in turn selects an edge. Player 1 starts by deleting an edge, and then player 2 fixes an edge (which has not been deleted yet); an edge fixed cannot be deleted later on by the other player. Player 2 wins if he succeeds in constructing a spanning tree of the graph; otherwise, player 1 wins.

The question is which graphs admit a winning strategy for player 1 (no matter what the other player does), and which admit a winning strategy for player 2.

Show that player 1 has a winning strategy if and only if G does not have two edge-disjoint spanning trees. Otherwise, player 2 has a winning strategy.