# Algorithms Final Exam

January 12, 2016,    09:10 - 12:10

Please answer the following 7 questions on the answer sheets provided. Be sure to write your name and student ID on all answer sheets you use. You may bring one A4-sized, hand-written, double-sided "cheat sheet". No other books, notes, or calculators may be used during the exam.

If you want to use any result or theorem that has been taught in class (including homeworks), you may do so but you must state the result or theorem clearly before using it.

## Problem 1 (15%)

Design a data structure that supports the following operations in amortized $O(\log n)$ time, where $n$ is the total number of elements. Briefly justify the correctness and analyze the running time.

Ins($k$): Insert a new element with key $k$
Extract-Median: Find and remove the element with the key being the median of all keys.
Union: Merge two different sets of elements

## Problem 2 (10%)

$n$ airports $A_1, A_2, \ldots, A_n$ are connected by $m$ flights. A traveler must start with one of the first $i$ airports $A_1, A_2, \ldots, A_i$ and wants to reach one of the last $j$ airports $A_{n-j+1}, A_{n-j+2}, \ldots, A_n$. His main goal is minimizing the number of flight connections. Design a $O(n+m)$-time algorithm to find the optimal route (or report that no such route exists). The running time must not depend on $i$ and $j$. Briefly justify the correctness and analyze the running time.

## Problem 3 (15%)

$n$ different courses are offered by NTUEE. Each course takes a full semester and has 0 to 3 other courses as prerequisites. Each course can only be taken after all prerequisites are completed and takes one semester to finish. Assume that all courses are offered every semester and a person can take any number of courses in one semester. Design an algorithm to find the minimum number of semesters required to finish all of these $n$ courses (given that you can complete all courses in some orderings). You algorithm must run in $O(n)$ time in order to receive full credit. Briefly explain the correctness and analyze the running time.

## Problem 4 (15%)

$n$ airports $A_1, A_2, \ldots, A_n$ are connected by some flights. The flight from $A_i$ to $A_j$ may be canceled due to weather conditions with probability $p_{ij}$. Given all flights between these airports and all cancellation probabilities, design an algorithm to find the route from $A_1$ to $A_n$ with the maximum probability of no flight cancellation at all. Briefly justify the correctness and analyze the running time. You must design an efficient algorithm for full credit, but any polynomial time algorithm gives partial credit.

For problems 5 and 6, you must either

1. design a polynomial time algorithm, briefly justify the correctness and analyze the running time, or
2. prove that the problem is NP-complete.

You may use the fact that SAT, 3-SAT, Set Cover, Independent Set, Hamiltonian Path and Hamiltonian Cycle problems are all NP-complete. Also, you do not need to prove that problems 5, 6 are in NP.

## Problem 5 (15%)

$m$ courses $C_1, C_2, \ldots, C_m$ will be offered in NTUEE in the next semester. Each course may require a different number of TAs. Assume that course $c_i$ requires $n_i$ TAs for every $i$ and every student can become a TA in at most two courses. Furthermore, only a subset $S_i$ of all students is qualified for being a TA in course $C_i$. Given all $n_i$ and $S_i$, the goal is to determine whether it is possible to assign enough TAs to all courses.

## Problem 6 (15%)

Given an undirected graph $G = (V, E)$ in which every edge $e$ has a non-negative cost $c_e$, a subset of vertices $V' \subseteq V$, and a constant $k$.

The goal is to determine if there exists a subset of edges $E' \subseteq E$ with total cost at most $k$ such that $E'$ connects all vertices in $V'$. (In other words, all vertices in $V'$ are in the same connected component of the graph $G' = (V, E')$.)

## Problem 7 (15%)

Given a connected undirected graph $G = (V, E)$ in which every edge $e$ has a edge cost $c_e$ satisfying $1 \leq c_e \leq 2$. Design an approximate algorithm to find the total cost of the minimum spanning tree. The output must be between $1 \cdot$ cost(MST) and $1.5 \cdot$ cost(MST). The running time of your algorithm must be $O(V + E)$. (Hint: An algorithm that directly outputs $2(|V| - 1)$ without even looking at the graph is a 2-approximate algorithm.)

---

Administrative issues:
1. The exam score and adjustment will be announced on 1/21(Thu) the latest.
2. You can check exam scores on 1/20(Wed) 10-12am at MD718.