CSE 421

Midterm review

Yin Tat Lee

Homework 3 Comments

Keep pseudocode concise and clear

- Avoid writing out the sorting algorithm (or any algorithm we taught).
  - This is difficult to read and may lead to more errors.
  - Better: **Sort jobs by $\boldsymbol{t_i / w_i}$ in ascending order**.
- Avoid long notation (such as jobs[i].w)
  - Better: w_i (Defined in the problem. So, you can just use it)

Proof

- Bad: My algorithm does X at each step, which brings me closer to a correct solution, therefore my algorithm is correct.
- Advice: Split the proof into multiple steps.
  - Example: To prove the algorithm find a certain subset X^*,
  - Split it to $X \subset X^*$ and $X^* \subset X$ where X is the output of the algorithm.
  - Example: To prove the algorithm compute something
  - Use induction and prove the k-step of the algorithm computes (something)

Format

Coverage: Lecture 1 – 12

Time: 50 min

Format:

18% true / false

27% fill in the blank

55% long question

- 1 question on graphs
- 1 question on greedy methods
- 1 question on divide and conquer

Only 1 question requires proof.

(Look at the sample midterms in the website)

Exercise for Midterm

5 minute to think

Given an undirected graph *G* with *n* vertices and *m* edges. Each edge represents a highway or a flight. Let c_e be the # hours needed to cross edge *e*.

Suppose that

- it takes 3 extra hours to pass through the security in airport.
- No extra hour for transferring from one flight to another.

Give a polynomial time algorithm to find the fastest way to go from vertex *s* to vertex *t*.
(Hints: Reduction.)

Exercise for Midterm

This midterm is easier this.

Given an undirected graph *G* with *n* vertices and *m* edges. Each edge represents a highway or a flight. Let c_e be the # hours needed to cross edge *e*.

Suppose that

- it takes 3 extra hours to pass through the security in airport.
- No extra hour for transferring from one flight to another.
- You cannot take more than 3 flights in the whole trip.

Give a polynomial time algorithm to find the fastest way to go from vertex *s* to vertex *t*.

(Hints: Reduction.)

Exercise for Midterm

Given a sequence of increasing integer $a_1, a_2, \cdots, a_n$. Assume there is i such that $a_i = i$. Give an algorithm to find such i in $O(\log n)$ time.

(Hints: Reduction.)

Exercise for Midterm

Given a complete binary tree with root r and n vertices. Give an algorithm to find k leaves of the tree in $O(k + \log n)$ time.

(Note: Every non-leaf of a complete binary tree has two children.)

Exercise for Midterm

Given a weighted directed acyclic graph with n vertices and m edges. Give an $O(n+m)$ time algorithm to find the shortest path distance from vertex s to all other vertices. (Hints: Topological sort.)

Exercise for Midterm

Given a connected graph with n vertices and m edges with $m \geq n$. Give an $O(n)$ time algorithm to find a cycle.

(Hints: Tree.)