

CS 344: Design and Analysis of Computer Algorithms

Rutgers: Spring 2022

## Practice Midterm Exam #2

Name: \_\_\_\_\_

NetID: \_\_\_\_\_

### Instructions

1. Do not forget to write your name and NetID above, and to sign Rutgers honor pledge below.
2. The exam contains 4 problems worth 100 points in total *plus* one extra credit problem worth 10 points.
3. The exam should be done **individually** and you are not allowed to discuss these questions with anyone else. This includes asking any questions or clarifications regarding the exam from other students or posting them publicly on Piazza (**any inquiry should be posted privately on Piazza**). You may however consult all the materials used in this course (video lectures, notes, textbook, etc.) while writing your solution, but **no other resources are allowed**.
4. Remember that you can leave a problem (or parts of it) entirely blank and receive 25% of the grade for that problem (or part). However, this should not discourage you from attempting a problem if you think you know how to approach it as you will receive partial credit more than 25% if you are on the right track. But keep in mind that if you simply do not know the answer, writing a very wrong answer may lead to 0% credit.

The only **exception** to this rule is the extra credit problem: you do not get any credit for leaving the extra credit problem blank, and there is almost no partial credit on that problem.

5. **You should always prove the correctness of your algorithm and analyze its runtime.** Also, as a general rule, avoid using complicated pseudo-code and instead explain your algorithm in English.
6. You may use any algorithm presented in the class or homeworks as a building block for your solutions.

---

### Rutgers honor pledge:

*On my honor, I have neither received nor given any unauthorized assistance on this examination.*

Signature: \_\_\_\_\_

Problem. #	Points	Score
1	25	
2	25	
3	25	
4	25	
5	+10	
Total	100 + 10	

**Problem 1.**

- (a) Suppose  $G = (V, E)$  is a directed graph with positive weight  $w_e$  on each edge  $e$ . Let

$$P = s, v_1, v_2, \dots, v_k, t$$

be a shortest path from a vertex  $s$  to a vertex  $t$  in  $G$ . Prove that for every  $i \in \{1, \dots, k\}$ , the path

$$P_i = s, v_1, \dots, v_i$$

is also a shortest path from  $s$  to  $v_i$ .

**(12.5 points)**

- (b) Suppose  $G = (V, E)$  is an undirected connected graph with positive weight  $w_e$  on each edge  $e$ . Prove that if the weight of some edge  $f$  is *strictly smaller* than weight of all other edges in  $G$ , then *every* minimum spanning tree (MST) of  $G$  contains the edge  $f$ .

(Note that to prove this statement, it is *not* enough to say that some specific algorithm for MST always picks this edge  $f$ ; you have to prove *all* MSTs of  $G$  contain this edge). **(12.5 points)**

**Problem 2.** Recall the job scheduling problem from the class: we have a collection of  $n$  processing jobs and the length of job  $i$ , i.e., the time to process job  $i$ , is given by  $L[i]$ . This time, you are given a number  $M$  and you are told that you should finish all your processing jobs between time 0 and  $M$ ; any job not fully processed in this window then should be paid a penalty that is the same across all the jobs. The goal is to find a schedule of the jobs that minimizes the penalty you have to pay, i.e., it minimizes the number of jobs not fully processed in the given window.

Design a greedy algorithm that given the array  $L[1 : n]$  of job lengths and integer  $M$ , finds the scheduling that minimizes the penalty in  $O(n \log n)$  time. **(25 points)**

A complete solution consists of three part: the algorithm (**10 points**), the proof of correctness (**10 points**), and the runtime analysis (**5 points**).

**Problem 3.** Crazy City consists of  $n$  houses and  $m$  bidirectional streets connecting these houses together, and there is always at least one way to go from any house to another one following these streets. For every street  $e$  in this city, the cost of maintaining this street is some positive integer  $c_e > 0$ . The mayor of Crazy City has come up with a brilliant cost saving plan: destroy(!) as many as the streets possible to maximize the cost of destroyed streets (so we no longer have to pay for their maintenance) while only ensuring that there is still a way for every house to reach mayor's house following the remaining streets.

Design an  $O(m \log m)$  time algorithm that outputs the set of streets with *maximum total cost* that should be destroyed by the mayor. **(25 points)**

A complete solution consists of three part: the algorithm or reduction (**10 points**), the proof of correctness (**10 points**), and the runtime analysis (**5 points**).

**Problem 4.** We have an undirected graph  $G = (V, E)$  with positive weight  $w_e > 0$  over each edge  $e \in E$ . We are additionally given two vertices  $s, t \in V$  and are promised that at least one of the shortest (minimum weight) paths from  $s$  to  $t$  uses at most 10 edges. Design and analyze an algorithm that in  $O(n + m)$  time finds a shortest path from  $s$  to  $t$  in  $G$  (the shortest path found by your algorithm does not necessarily need to be the one that uses at most 10 edges). **(25 points)**

A complete solution consists of three part: the algorithm or reduction (**10 points**), the proof of correctness (**10 points**), and the runtime analysis (**5 points**).

**Problem 5.** [Extra credit] Design and analyze an algorithm that given an undirected graph  $G = (V, E)$ , in  $O(n + m)$  time determines whether or not  $G$  is *bipartite*. Recall that a graph is bipartite if its vertices can be partitioned into two sets  $L$  and  $R$  such that all edges of the graph are between  $L$  and  $R$ . (+10 points)

## Extra Workspace