ShanghaiTech University Final Examination Cover Sheet

Academic Year : 2022 Term: Spring

Course-offering School: SIST

Instructor: Rui Fan

Course Name: Algorithm Design and Analysis / 算法设计与分析

Course Number: CS 240

General Instructions:

1. All examination rules must be strictly observed throughout the entire test. Any form of academic dishonesty is strictly prohibited, and violations may result in severe sanctions.

2. Based on government policy, you will be videorecorded during the entire examination, and the recordings will be provided to the government for possible inspection.

3. This exam is closed book. You may only use 2 pages of notes. You may not use your computer or phone for any purposes other than downloading the exam, submitting your solutions and for videorecording.

Exam Instructions:

1. In all problems in which you are asked to design algorithms, you should clearly describe how your algorithm works, and provide pseudocode if this improves clarity. You need to argue why your algorithm is correct, and analyze your algorithm when asked to.

2. All answers must be written neatly and legibly in English.

1. Answer true or false for the following.

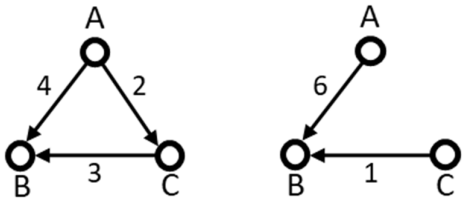
   a. Let f be a maximum $s-t$ flow on a graph $G=(V,E)$, and let B be the set of vertices in V which can reach t in the residual graph G_f. Then $(V-B,B)$ is a minimum capacity $s-t$ cut.
   b. Let f be a maximum $s-t$ flow on a graph $G=(V,E)$, and let (S,T) be the corresponding minimum capacity $s-t$ cut. Then every edge e which crosses between S and T has flow $f(e)$ equal to its capacity $c(e)$.
   c. If a problem B is NP-hard, and problem $A \leq_P B$, then A is NP-hard.
   d. If a problem $A \in P$, then $A \leq_P B$ for every problem $B \in NP$.
   e. Let $T(n)$ be the complexity of a problem on input size n, and suppose $T(n) = T\left(\frac{9n}{10}\right) + T\left(\frac{3n}{20}\right) + O(n)$. Then $T(n) = O(n)$.

   (10 points, 2 points each)

2. Given a sequence of numbers, you want to find a subsequence whose values are all increasing, and whose sum is maximum. For example, given the sequence $[8,4,15,7,14,12]$, the optimal subsequence is $[4,7,14]$, which has sum 25. Design an efficient algorithm for this problem, and analyze its complexity.

   (10 points)

3. You and a group of friends have borrowed money from each other, and now it's time for everyone to pay what they owe. Your goal is to do this while mimimizing the total amount of money transferred. For example, in the left figure below, each edge (u,v) indicates that u owes v the number shown on the edge. If each person pays all the amounts indicated in the left figure, the total amount of money transferred is 9 RMB. However, the right figure shows an equivalent way for everyone to get their money, which only transfers a total of 7 RMB, and this amount is optimal. Design an efficient algorithm to minimize the total amount of money transferred, argue why your algorithm is correct, and analyze the complexity of your algorithm.



   (10 points)

4. Suppose you are given an array of n numbers and want to find the k'th largest number, for some $1 \leq k \leq n$. There are deterministic algorithms to solve this problem in $O(n)$ time, but they are somewhat complicated. Design a simple randomized algorithm to solve this problem in $O(n)$ expected time. Clearly describe the algorithm and analyze its expected time complexity.

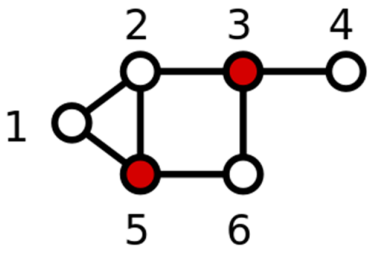
   Hint: Use an approach similar to randomized Quicksort.

   (10 points)

5. Recall from class that we found a 2-approximation algorithm for the k-center problem. You will now prove this is essentially optimal, by showing that there does not exist a polynomial time $(2-\epsilon)$-approximation algorithm for k-center, for any $\epsilon > 0$, unless P = NP.

   To do this, we first consider k-center as a graph problem. Let G be a *complete*, undirected, weighted graph on vertex set V, where the weights satisfy the triangle inequality. That is, for any $u, v, w \in V$, there exist edges $(u,v), (u,w), (v,w)$ in G, and $d(u,v) \leq d(u,w) + d(w,v)$, where $d(e)$ denotes the weight of an edge e. For any node $v \in V$ and set $S \subseteq V$, define $d(v,S) = \min_{u \in S} d(u,v)$. The k-center problem is to find a set of k nodes $S \subseteq V$ such that $\max_{v \in V} d(v,S)$ is the minimum, among all possible choices for S. Let $d^* = \max_{v \in V} d(v,S)$ be the value for this optimal set S.

   Next, we define the *k-dominating set* problem. Here, we are given an undirected, unweighted, and not necessarily complete graph $G = (V,E)$, and need to determine whether there exists a set of k nodes $S \subseteq V$ such that $\forall v \in V \exists u \in S: (u,v) \in E$. That is, we want to know if there are k nodes such that every node in V is adjacent to one of the k nodes. For example, the red nodes in the graph below are a 2-dominating set, and this graph does not have a 1-dominating set.



   It is known that the k-dominating set problem is NP-complete (when k is part of the input). We use this to prove k-center is NP-hard to approximate to factor $2-\epsilon$. In particular, suppose for contradiction there exists a polytime $(2-\epsilon)$-approximation algorithm A for k-center, for some $\epsilon > 0$. That is, for any instance of k-center, if the optimal solution value is d^*, A returns a value $\tilde{d} \leq (2-\epsilon)d^*$. Then, given an instance G of k-dominating set, describe how to construct in polytime an instance

G' of k-center based on G, so that by running A on G', you can determine whether G has a solution in polytime. Note that you need to define the vertex set of G' and the weights of all its edges, and also describe how the output of A tells you whether G is solvable.

Hint: It suffices to make all the weights in G' be either 1 or 2.

(10 points)

END OF EXAM