Name: ____________________

Student ID: ____________________

School: ____________________

Year of Entrance: ____________________

ShanghaiTech University Final Examination Cover Sheet

Academic Year : 2021 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

Exam Instructions for Students:

1. All examination rules must be strictly observed throughout the entire test, and any form of cheating is prohibited.

2. Other than allowable materials, students taking closed-book tests must place their books, notes, tablets and any other electronic devices in places designated by the examiners.

3. Students taking open-book tests may use allowable materials authorized by the examiners. They must complete the exam independently without discussion with each other or exchange of materials.

For Marker's Use:

Section	1	2	3	4	5	6					Total
Marks											
Recheck											

Marker's Signature:
Date:

Rechecker's Signature:
Date:

Instructions for Examiners:

1. The format of the exam papers and answer sheets shall be determined by the school and examiners according to actual needs. All pages should be marked by the page numbers in order (except the cover page). All text should be legible, visually comfortable and easy to bind on the left side. A4 double-sided printing is recommended for the convenience of archiving (There are all-in-one printers in the university).

2.The examiners should make sure that exam questions are correct and appropriate, If errors are found in exam questions during the exam, the examiners should be responsible to respond on site, which will be taking into account in the teaching evaluation.

Instructions for Students

In all problems in which you are asked to design algorithms, you should clearly describe how your algorithm works, provide code or pseudocode when asked to, and argue why your algorithm is correct.

All answers must be written neatly and legibly in English.

1. Suppose you are given an $n \times n$ matrix of numbers, and want to determine whether a particular number x is in the matrix. Assume the matrix is sorted in increasing order both along every row and along every column. An example of such a matrix is shown below. Design an efficient algorithm for this problem and analyze its complexity.

$$\begin{pmatrix} 2 & 3 & 5 \\ 3 & 4 & 6 \\ 5 & 7 & 8 \end{pmatrix}$$

(10 points)

2. A *celebrity* is a person who is known by everyone, but who does not know anyone else. Given a group of n people, your goal is to determine whether the group contains a celebrity or not (notice the group contains either 0 or 1 celebrity). To do so, you can take a pair of people x and y, and ask if x knows y, getting a yes/no answer in response. Design an efficent algorithm which minimizes the number of questions you ask, and analyze your algorithm's complexity (i.e. number of questions asked).

(10 points)

3. Given a set S of integers, let $h(S) = \frac{\sum_{x \in S} x}{|S|}$ be the average value in S. The AVERAGE-SUM problem asks whether there exists a subset of S which sums to $h(S)$. Note that this problem is similar to the SUBSET-SUM problem, except that the target value is always the average value of S. Show that AVERAGE-SUM is NP-Complete.

(10 points)

4. Consider an algorithm which takes as input n numbers, and outputs the smallest $n/2$ numbers in sorted order. In addition, the algorithm is only allowed to make comparisons between a pair of numbers. Show that the algorithm must make $\Omega(n \log n)$ comparisons.

   Hint: Recall that any comparison-based algorithm which sorts n numbers needs to make $\Omega(n \log n)$ comparisons.

   (10 points)

5. Suppose you have a fair coin which generates a random bit each time you flip it. Design an algorithm which uses this coin to select a value uniformly at random from the set $\{1, \dots, n\}$, for any n. Your algorithm should use $O(\log n)$ coin flips in expectation.

(10 points)

6. In the MAX-k-CUT problem, we are given an undirected graph $G = (V, E)$ and an integer k. We want to partition V into k parts $V_1, \dots, V_k$, such that the number of edges crossing between different parts is maximized. That is, we want to maximize $|\{(u, v) \in E \mid u \in V_i, v \in V_j \text{ for some } i \neq j\}|$. This problem generalizes the MAX-CUT problem. Design a $\frac{k-1}{k}$-approximation algorithm for MAX-k-CUT, and prove that your algorithm is correct.

(10 points)

END OF EXAM