CS 240 Algorithm Design and Analysis (Spring 2019)
Final Exam

Name (in Chinese): _____

ID#: _____

Instructions

- Time: 1:00-2:40pm (100 minutes)
- This exam is closed-book, but you may bring an A4-size cheat sheet. Put all the study materials and electronic devices into your bag and put your bag in the front, back, or sides of the classroom.
- You can write your answers in either English or Chinese. You can use both sides of the paper.
- Two blank pieces of paper are attached on the back, which you can use as scratch paper. Raise your hand if you need more paper.

1 (10 pt)

Each question has one or more correct answer(s). Select all the correct answer(s). For each question, you get 0 point if you select one or more wrong answers, but you get 1 point if you select a non-empty proper subset of the correct answers.

1	2	3	4	5

1. Which of the following is correct?

   A. $n^{\sqrt{n}}$ is $O(2^n)$

   B. 100^{100} is $O(\log n)$

   C. $(\log n)^n$ is $O(n^{\log n})$

   D. n^2 is $O((\log n)^{\log n})$

   E. None of above

2. Which of the following is known to be correct?

   A. 3-SAT $\in$ P

   B. 3-SAT $\in$ NP

   C. 3-SAT $\in$ NP-complete

   D. 3-SAT $\in$ NP-hard

   E. None of the above.

3. Which of the following is known to be correct?

   A. $X \in \text{P} \Rightarrow X$ has a poly-time certifier
   B. $X \in \text{NP} \Rightarrow X$ has a poly-space certifier
   C. $X \in \text{PSPACE} \Rightarrow X$ does not have a poly-time certifier
   D. $X \in \text{PSPACE-complete} \Rightarrow X$ does not have a poly-time certifier
   E. None of above

4. Which of the following is known to be correct?

   A. 2-COLOR $\leq_p$ 3-COLOR
   B. 2D-Matching $\leq_p$ 3D-Matching
   C. 2D-Matching $\leq_p$ 2-COLOR
   D. 3D-Matching $\leq_p$ 3-COLOR
   E. None of the above.

5. Which of the following is known to be correct?

   A. If $X \in \text{NP} \cap \text{co-NP}$, then $X \in \text{P}$
   B. If $X \in \text{NP}$, then its complement $\bar{X} \in \text{co-NP}$
   C. If $X \notin \text{NP}$, then its complement $\bar{X} \notin \text{co-NP}$
   D. If $\text{P} = \text{PSPACE}$, then $\text{co-NP} = \text{NP}$
   E. None of the above.

2 (10 pt)

We define a sequence of matrices $H_1, H_2, \ldots, H_k$ as follows.

1. $H_0 = [1]$
2. For $k > 0$, $H_k = \begin{bmatrix} H_{k-1} & H_{k-1} \\ H_{k-1} & -H_{k-1} \end{bmatrix}$

Let v be a vector of length $n = 2^k$. Computing the product $H_k v$ in the usual way would take $O(n^2)$ arithmetic operations (additions, subtractions, multiplications). Describe an algorithm that can compute the product with fewer operations and write down the time complexity of your algorithm.

3 (10 pt)

Students from several universities gather together for a social event. To increase social interaction, the event organizers want to assign these students to meeting rooms such that no two students from the same university are in the same room. Assume that there are p universities and there are a_i students from the i-th university. Also assume that there are q rooms and the j-th room can accommodate at most b_j students. Design a polynomial-time algorithm that outputs a valid room assignment, or returns an error message if no such assignment exists.

4 (10 pt)

Suppose a cashier needs to make change for v RMB, and has available bills of denominations $x_1, ..., x_k$, where $1 \leq x_1 < ... < x_k$ are positive integers. Give an algorithm to compute the number ways to make change, ignoring the order of the bills. For example, when $v = 11$ and there are 3 types of bills of denominations $1, 5$ and 10, there are 4 ways to make change, namely $\{10, 1\}, \{5, 5, 1\}, \{5, 1, 1, 1, 1, 1, 1\}$ and $\{1, \ldots, 1\}$ with 11 1's.

What is the running time of your algorithm?

5 (10 pt)

Recall from class that there is an efficient 3/2 approximation algorithm for the *metric* TSP problem, where for any three nodes i, j and k, we have $w_{i,j} + w_{j,k} \geq w_{i,k}$ ($w_{u,v}$ denotes the weight of the edge from node u to v). Now, consider the general TSP problem, where the edges are allowed to have *arbitrary* weights. Prove that for any constant $k \geq 1$, there is no polynomial time k-approximation algorithm for general TSP, unless $P = NP$.

Hint: Recall that finding a Hamiltonian Cycle in a graph is NP-complete. Construct a graph which includes some edges with very large weights, such that finding an approximate TSP in this graph allows finding a Hamiltonian Cycle in another graph.

6 (10 pt)

Suppose we are given three $n \times n$ matrices A, B and C, and we want to check whether $AB = C$. One method is to directly multiply A and B and compare the result to C. This would take about $\Omega(n^{2.37})$ time using the fastest known matrix multiplication algorithm. However, since we only need to check whether $AB = C$, we can actually do better.

Give a Monte Carlo randomized algorithm which correctly decides whether $AB = C$ with $\Omega(1)$ probability, and runs in $O(n^2)$ time. Argue why your algorithm succeeds with $\Omega(1)$ probability and runs in $O(n^2)$ time.

Hint: Consider multiplying both sides of $AB = C$ by a vector. How much time does this take? What does the result tell you?