Spring 2013 **CMSC 351: Midterm** Hamid Mahini
Time: 1 Hour, 15 Minutes

Type A

WAIT FOR INSTRUCTIONS BEFORE BEGINNING

HONOR PLEDGE: "I pledge on my honor that I have not given or received any unauthorized assistance on this examination."

Signature and UID: ____

Print name: ____

- ***Write your answers with enough detail about your approach and concepts used, so that the grader will be able to understand it easily. You should PROVE the correctness of your algorithms either directly or by referring to a proof in the book.***
- ***The sum of the grades is 105, but your grades would be out of 100 (thus you get 5 bonus points by solving all the problems).***
- ***Select the best choice for the first 5 problems and mark it by X in the table below***

Problem	1	2	3	4	5
A					
B					
C					
D					
E					

DO NOT WRITE BELOW THIS LINE

Problem 1-5:	
Problem 6:	
Problem 7:	
Problem 8:	
Problem 9:	
TOTAL:	

Multiple-choice Problems (Answer ONLY in the TABLE in the FIRST PAGE and NOT HERE):

Problem 1. (5 points) $\sum_{i=1}^{n} i^3$ is

a) $\Theta(n^2)$ b) $\Theta(n^3)$ **c) $\boldsymbol{\Theta(n^4)}$** d) $O(n^3)$ e) $O(n^3 \log n)$

Problem 2. (5 points) Let T(n)=9T($\left\lfloor \frac{n}{4} \right\rfloor$)+ n^2-4n and T(0)=1. Then T(n) is

a) $\Theta(n^2 \log n)$ **b) $\boldsymbol{\Theta(n^2)}$** c) $\Theta(n \log n)$
d) $\Theta(n \log^2 n)$ e) $\Theta(n^{\log_4^9})$

Problem 3. (5 points) Let T(n)=4T($\lfloor \sqrt{n} \rfloor$)+5 and T(0)=1. Then T(n) is

a) $\Theta(\log n)$ b) $\Theta(\log(\log(n)))$ c) $\Theta(\sqrt{n})$
d) $\boldsymbol{\Theta(\log^2 n)}$ e) $\Theta(n)$

Problem 4. (5 points) How many times do we call the procedure Proc() below?

for *i:=1* to *n* do

for *j:=i* to *n* do

for *k:= i* to *n* do

Proc(i, j, k);

a) n(n-1)(n-2)/6 b) $n^3 - n^2$ c) $O(n^2)$ **d) $\boldsymbol{O(n^3)}$** e) $n^3/2$

Problem 5. (5 points) What is A[5] if we add number 17 to the following heap?

Index	1	2	3	4	5	6	7	8	9	10
A	20	15	11	8	2	10	9	3	4	1

a) 8 b) 11 **c) 15** d) 17 e) 10

Regular Problems:

Problem 6. (20 points) Consider the following functions:

(func 0) 100
(func 1) $\log(n^3)$
(func 2) $2^n n$
(func 3) $200n^{0.01}$
(func 4) 3^n
(func 5) $\sqrt{4n}$

Complete the table below. In the entry at row "$func\ i$" and column "$func\ j$", cross the best relation you can prove between "$func\ i$" and "$func\ j$", i.e. put ONLY ONE of O or Ω or Θ. (You do not need to write the proofs).

Functions	$func$ 0	$func$ 1	$func$ 2	$func$ 3	$func$ 4	$func$ 5
$func$ 0	Θ	O	O	O	O	O
$func$ 1	Ω	Θ	O	O	O	O
$func$ 2	Ω	Ω	Θ	Ω	O	Ω
$func$ 3	Ω	Ω	O	Θ	O	O
$func$ 4	Ω	Ω	Ω	Ω	Θ	Ω
$func$ 5	Ω	Ω	O	Ω	O	Θ

Problem 7. (20 points) Prove by induction that $\prod_{i=2}^{n}(1-\frac{1}{i^2}) = \frac{n+1}{2n}$ for every $n \geq 2$.

Let $P(n)$: $\prod_{i=2}^{n}(1-\frac{1}{i^2}) = \frac{n+1}{2n}$

Base Case: For n=2 we have $P(2)$: $1 - \frac{1}{4} = \frac{3}{4} = \frac{2+1}{2.2}$

Induction Hypothesis: For all $k < n$ we have $\prod_{i=2}^{k}(1-\frac{1}{i^2}) = \frac{k+1}{2k}$

Inductive Step: For n we have

$$\prod_{i=2}^{n}(1-\frac{1}{i^2}) = \prod_{i=2}^{n-1}(1-\frac{1}{i^2}) \times (1-\frac{1}{n^2}) \qquad \text{By simplification}$$

$$= \frac{n}{2(n-1)} \times (1-\frac{1}{n^2}) \qquad \text{By Induction Hypothesis}$$

$$= \frac{n}{2(n-1)} \times (\frac{n^2-1}{n^2})$$

$$= \frac{n}{2(n-1)} \times (\frac{(n-1)(n+1)}{n^2})$$

$$= \frac{n+1}{2n}$$

Problem 8. (20 points) Given two sets S_1 and S_2, and a real number x. Find whether there exists an element from S_1 and an element from S_2 whose sum is exactly x. The algorithm should run in time $O(n \log n)$, where n is the total number of elements in both sets.

First we sort the elements of S_1 using merge sort. Because the total number of elements in S_1 is at most n this step takes $O(n\ log\ n)$. Then, for each element e in S_2 we binary search for the value $x - e$ in the sorted set S_1. If there exists an element e in S_2 where $x - e$ is in S_1 then we have found the two elements whose sum is x. Each binary search takes $O(\log n)$ and we run at most n (number of elements in S_2) binary searches. Hence the running time for the second step is $O(n \log n)$. Therefore the total running time is $O(n \log n)$.

Problem 9. (20 points) Explain the mergesort algorithm in details and analyze its running time. It is not necessary to write pseudo-code.

See pages 130 and 131 of the book.