University of Waterloo
School of Computer Science
CS 341 Algorithms, Winter, 2017
2 Hour Sample Midterm Exam
February 17, 2017

Instructors: Semih Salihoglu, Doug Stinson, Yaoliang Yu

Name:..

Student ID:..

Lecture Section 2 & 4(Stinson) 1 & 3 (Salihoglu) 5 (Yu)

*** Circle section in which you wish to pick up your midterm***

Question	Marks	Total
1		10
2		15
3		15
4		18
5		22
6		20
Total		100
Verified		100

Instructions

- NO CALCULATORS OR OTHER AIDS ARE ALLOWED.
- You should have 13 pages in total.
- Make sure your name and student ID is recorded on the first page.
- Solutions will be marked for clarity, conciseness and correctness.
- If you need more space to complete an answer, you may continue on the two blank pages at the end.
- The backs of pages can be used for rough work, and will not be marked unless you specifically indicate you wish them considered.

Useful Facts and Formulas

1. Master Theorem (simplified version)
   Suppose that $a \geq 1$ and $b > 1$. Consider the recurrence
   $$T(n) = a\,T\left(\frac{n}{b}\right) + \Theta(n^y)$$
   in sloppy or exact form. Denote $x = \log_b a$. Then
   $$T(n) \in \begin{cases} \Theta(n^x) & \text{if } y < x \\ \Theta(n^x \log n) & \text{if } y = x \\ \Theta(n^y) & \text{if } y > x. \end{cases}$$

2. Master Theorem (general version)
   Suppose that $a \geq 1$ and $b > 1$. Consider the recurrence
   $$T(n) = a\,T\left(\frac{n}{b}\right) + f(n)$$
   in sloppy or exact form. Denote $x = \log_b a$. Then
   $$T(n) \in \begin{cases} \Theta(n^x) & \text{if } f(n) \in O(n^{x-\epsilon}) \text{ for some } \epsilon > 0 \\ \Theta(n^x \log n) & \text{if } f(n) \in \Theta(n^x) \\ \Theta(f(n)) & \text{if } f(n)/n^{x+\epsilon} \text{ is an increasing function of } n \\ & \text{for some } \epsilon > 0. \end{cases}$$

3. $a^{\log_b c} = c^{\log_b a}$

4. $\sqrt{5} \approx 2.23, \qquad \log_2 3 \approx 1.58, \qquad \pi \approx 3.14$

1. *[10 marks total]* For each question below, give your answer together with a brief explanation. Show computations if it is appropriate to do so.

   (a) *[4 marks]* True or false: the closest pair problem *in one dimension* (i.e., for a set of points all on a line) can be solved in $O(n \log n)$ time without using the divide-and-conquer method from class. Briefly explain your answer.

   (b) *[3 marks]* Give a simplified Θ-bound for the expression $57n^{\sqrt{5}} + 39\sqrt{n}\, 3^{\log_2 n}$.

   (c) *[3 marks]* Which of the following two functions has the higher growth rate: $2^{\pi \log_2 n}$ or $n^3(\log_2 n)^{20}$?

2. *[15 marks] Recurrences.*
Solve the following recurrence by using the recursion-tree method (you may assume that n is a power of 8):

$$T(n) = \begin{cases} 4T(n/8) + n^2 & \text{if } n > 1 \\ 2 & \text{if } n = 1. \end{cases}$$

Express the answer exactly as a sum, and then determine the growth rate of $T(n)$.

3. *[15 marks] Pseudocode analysis.*
Give a detailed analysis of the complexity of the procedure $f(n)$ in terms of the input parameter n. You can assume that n is a power of 2 in order to simplify the analysis.

```
Procedure f(n)
i ← 1
S ← 0
for j ← 1 to n do
    S ← S + j^3
m ← n
while m ≥ 1 do
    for j ← 1 to m do
        S ← S + (i − j)^2
    m ← ⌊m/2⌋
    i ← i + 1
print(S)
```

4. *[18 marks total] Greedy algorithms.*

In the *Interval covering* problem, we are given a set X of n distinct real numbers $X = \{x_1, \ldots, x_n\}$, and an *interval length* L. We are required to find the minimum possible number of closed intervals, each of length L, such that every x_i is contained in at least one of the intervals. That is, we wish to find m intervals, say $I_1 = [a_1, a_1 + L], \ldots, I_m = [a_m, a_m + L]$, whose union contains all the x_i's, with m as small as possible.

In this question, we consider two possible greedy strategies for this problem.

> Strategy 1: Choose an interval that covers the maximum number of elements in X; remove all elements covered by this interval; and repeat.
>
> Strategy 2: Let x be the smallest element in X; choose the interval $[x, x + L]$; remove all elements covered by this interval; and repeat.

(a) *[6 marks]* By considering the problem instance $n = 6$, $X = \{2, 9, 12, 14, 17, 24\}$, and $L = 10$, prove that one of the two given strategies does not always find the optimal solution to the *Interval covering* problem.

(b) *[12 marks]* Give a complete proof that the other strategy always finds an optimal solution to the *Interval covering* problem.

5. *[22 marks] Divide-and-conquer.*
Define the following sequence of numbers: $F_0 = 0$, $F_1 = 1$, and

$$\begin{aligned} F_{2n} &= (F_n + F_{n-1})^2 - {F_{n-1}}^2 \\ F_{2n+1} &= (F_n + F_{n-1})^2 + {F_n}^2 \end{aligned}$$

(This is in fact the Fibonacci number sequence.)

(a) *[10 marks]* Give a pseudocode description of an efficient recursive algorithm to compute F_n for a given integer $n \geq 0$, based on the above definition.

(b) *[12 marks]* Determine (using O notation) the running time of your algorithm by writing a recurrence and solving it with the master method. Here, runnning time is measured in terms of the number of bit operations. Assume that the multiplication of two k-bit numbers requires $O(k^{1.59})$ time by Karatsuba's algorithm. You may use the fact that $F_n \leq 2^n$, so the number of bits in F_n is at most n (but mention where this is used in your analysis).

(Extra space.)

(Extra space.)