COMPSCI 320SC 2013 Midterm Test

Attempt *all* questions. (Use of calculators is NOT permitted.)

Put the answers in the space below the questions. Write clearly and *show all your work*!

Marks for each question are shown below and just before each answer area.

This 50 minute test is worth 10% of your final grade for the course.

Question #:	1	2	3	4	Total
Possible marks:	5	5	5	5	20
Awarded marks:					

University ID: ____________________________

Student Name: ____________________________

Student Signature: ____________________________

Time Finished: ____________________________

1. Recall the street lights problem from your first programming assignment. Your are given a road length n and a set of m lights with non-negative integer R-values, where 0 means that the bulb can only light up a section outside one house, 1 means lighting a given house plus two immediately neighboring houses from the lamp pole, and an R-value of k means that houses a distance k away, in addition, can be illuminated.

   (a) If we have a set of five light bulbs of R-values $\{3, 5, 2, 6, 1\}$, what is the minumum number of bulbs needed to light the street of length $n = 24$? (***3 marks***)

   (b) Give a formal proof that the greedy algorithm that sorts the set of bulbs in decreasing order (and picks bulbs until all of the street is lit) is optimal. Prove this for any input, not just the one example given in part (a). (***2 marks***)

2. Consider the following divide-and-conquer "algorithm":

function printer(int n)
    for $i = 1$ **to** n **do**
        for $j = 1$ **to** $\sqrt{n}$ **do**
            printline "hello world"
    if $n > 0$ **then**
        for $k = 1$ **to** 5 **do**
            printer($\lfloor n/3 \rfloor$)

Let $T(n)$ denote the number of lines of output generated by a call of printer(n).

(a) Provide a recurrence equation for $T(n)$. (***3 marks***)

(b) Solve the recurrence asymptotically for general n.

You may want to make use of the following 'master recurrence theorem':
Assume $t(n) = a \cdot t(n/b) + g(n)$, where $g(n) \in \Theta(n^c)$, is the total time for a divide-and-conquer algorithm. Then:

$$t(n) \in \begin{cases} \Theta(n^c) & \text{if } a < b^c \\ \Theta(n^c \log n) & \text{if } a = b^c \\ \Theta(n^{\log_b a}) & \text{if } a > b^c \end{cases}$$

(***2 marks***)

3. Consider the *24 Hour Interval Scheduling Problem*. This is the same as the Interval Scheduling Problem studied in class, except requested jobs (of length at most 24 hours) may start one day and finish the next day. Given a set of jobs (start, stop) clock times, we want to find as many jobs as possible that can be run during each day (any 24-hour period).

   (a) Consider the following four jobs, given in 24 hour times, (18:00,06:30), (21:00,04:00), (03:00,14:00) and (13:00,19:00). What is an optimal solution of non-overlapping jobs? How many different optimal solutions are there in total? (***3 marks***)

   (b) Give a polynomial-time greedy algorithm that solves this problem (for arbitrary input). Briefly justify the correctness. (***2 marks***)

4. Recall that we covered two divide-and-conquer algorithms to finding the k-th smallest from a list of n integers. One was `QuickSelect` (Hoare's selection), detailed in CS220, and the other was `Median-of-Medians` (by Blum, Floyd, Pratt, Rivest and Tarjan in 1973) from CS320.

   (a) What is the average case and worst case complexity of `QuickSelect`? (***2 marks***)

   (b) What is the worst case and average case complexity of `Median-of-Medians`? (***2 marks***)

   (c) Explain how one can use a heap data structure to find the k-th smallest in expected time $O(n + k \log n)$. (***1 marks***)