COMPSCI 320SC 2005 Midterm Exam

Attempt *all* questions. Put the answers in the space below the questions. Write clearly! You may continue your answers onto the "overflow" page provided at the end of the test, if necessary.

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

Use of calculators is NOT permitted.

This one hour (60 minutes) test is worth 15% of your final grade for the course.

Question #:	1	2	3	4	5	6	Total
Possible marks:	10	10	10	10	10	10	60
Awarded marks:							

University ID: _____

Student Name: _____

Student Signature: _____

Time Finished: _____

1. Suppose that I have a new algorithm for multiplication of integers that works by dividing integers x, y of size n into three parts of as equal size as possible, and computing the product xy by means of 5 multiplications of the parts, plus a fixed number of additions and shifts.

   Let $g(n)$ be the total worst-case time taken to perform the additions, shifts and overhead.

   (a) Write down a recurrence for the worst-case running time of this algorithm on an instance of size n. (***5 marks***)

   (b) Is this algorithm likely to be better than the primary school multiplication algorithm when used on large integers? Give full explanation. (***5 marks***)

2. Consider the mergesort recurrence

$$t(n) = \begin{cases} 0 & \text{if } n = 1; \\ t(\lceil n/2 \rceil) + t(\lfloor n/2 \rfloor) + n - 1 & \text{if } n > 1. \end{cases}$$

(a) What are the values of $t(3), t(4), t(5)$? (***3 marks***)

(b) Solve the recurrence exactly assuming that n is a power of 2. Express your answer in terms of a simple formula involving n only. (***5 marks***)

(c) Solve the recurrence asymptotically for general n. Justify your answer. (***2 marks***)

3. This question concerns certain divide-and-conquer algorithms covered in class. Recall that the *selection problem* is: given a list L of n integers and an integer k with $1 \leq k \leq n$, find the k-th smallest element of L. If $k = \lceil n/2 \rceil$, we call this the problem of finding the median.

   (a) Suppose that we have an algorithm that finds the median of a list of integers in worst-case linear time. Explain how we can use this as a subroutine in a divide-and-conquer algorithm to solve the general selection problem in worst-case linear time. Give full details. (***6 marks***)

   (b) Why is the procedure of (ii) not performed in practice? What is more usually done? What are the advantages and disadvantages of each method? (***4 marks***)

4. (a) Assume that the input size of a problem is $m = n^2$ and the algorithm runs in time $O(n^3)$. What is the running time in terms of m? Justify your answer. (***3 marks***)

   (b) Assume that the input size of a problem is $m = n \lg n$ and the algorithm runs in time $O(n^n)$. What is the running time in terms of m? Justify your answer. (***3 marks***)

   (c) Suppose we have a connected edge-weighted graph with n vertices and n edges. Each edge weight can be stored in a 32-bit integer. Give an algorithm that runs in time $O(n)$ that will find a minimum spanning tree. Explain the running times of all data structure methods you use when proving your linear-time algorithm. (***4 marks***)

5. Let $P_1, P_2, \ldots, P_n$ be programs to be stored on a disk. Program P_i requires s_i kilobytes of storage, and the capacity of the disk is D kilobytes, where $D < \Sigma_{i=1}^{n} s_i$. Prove that the greedy algorithm that selects programs in order of nondecreasing s_i will always maximize the number of programs that can be stored on the disk. (***10 marks***)

6. Consider the following word formatting problem. Input is a sequence of n words of lengths $w_1, w_2, \ldots, w_n$ and a line column width W to right justify the text. (Assume $W \geq w_i$ for all $1 \leq i \leq n$ and that each word does not contain any spaces but may have punctuation marks included.) We wrap the sequence of words on as many lines as needed provided the sum of the word lengths is at most W when allowing for exactly one space gap between each word. Let the "goodness" of a valid format be the sum of the squares of the extra filler space needed to pad all lines (except the last line) to width W.

   (a) With $W = 10$, give an example of 10 words such that the natural greedy algorithm does not give the "best" valid format. (***3 marks***)

   (b) Give a dynamic program that solves this problem efficiently. Please state formally what you are considering as subproblems. (***5 marks***)

   (c) Calculate and justify the running time of your algorithm. (***2 marks***)