Faculty of Science
Final Examination

Computer Science 308-250B
Introduction to Computer Science

Examiner: Prof. Claude Crépeau **Date:** April 15, 2004
Associate Examiner: Lecturer John M. Mercer **Time:** 14:00 – 17:00

INSTRUCTIONS:
- This examination is worth 60% of your final grade.
- The total of all questions is 65 points.
- Each question is assigned a value found in brackets next to it.
- O P E N • B O O K S •/• O P E N • N O T E S
- Faculty standard calculator permitted only.
- This examination consists of 4 pages including title page.
- This examination consists of 8 questions.

SUGGESTION : read all the questions and their values before you start.

[9pts]

1) Consider the following list of numbers : { 15, 9, 1, 13, 17, 21, 3, 7 }.

- Put these numbers in a binary search tree with largest possible height.
- Put these numbers in a binary search tree with smallest possible height.
- Put these numbers in a min-HEAP.
- In each case tell if the solution is unique or not and why.

[8pts]

2) Suppose you are given a (single) queue Q containing distinct integers.

- A) Find an algorithm (in pseudo-code) to invert the content of the queue using queue operations on Q and nothing else than a fixed number of extra **int** variables (no array, no stack, no other queue). Example:

  { 15, 9, 1, 13, 17, 21, 3, 7 } becomes { 7, 3, 21, 17, 13, 1, 9, 15 }.

- B) Estimate the worst-case running time of your algorithm and express this time function using the Big-O notation.

[+5pts]

- C) Argue that this would be impossible with a single stack S instead of a queue.

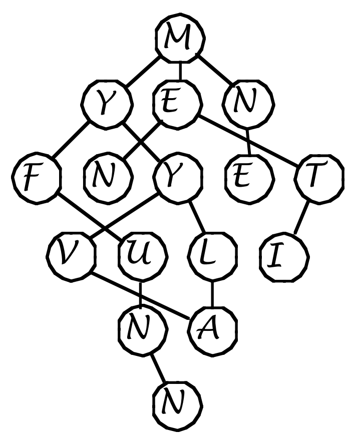
[9pts]

3) Consider the following graph. List the sequence of nodes visited according to a

a)

• Depth-first search starting at (M).
(DFS chooses neighbors in the order "left to right")

• Breath-first search starting at (M).
(BFS chooses neighbors in the order "left to right")



b) Re-label the vertices of this graph so that a BFS starting at (M) will produce the string "MY FUNNY VALENTINE". (BFS chooses neighbors "from left to right")
(Write your answer on 6 lines; the current graph labeling would be:

M
YEN
FNYET
VULI
NA
N

[8pts]

4) Let H_n be the number of <u>distinct</u> heaps that can be built with n <u>distinct</u> keys. By the definitional properties of a heap we may assert that H_n may be defined as

$$H_n = \begin{cases} 1 & \text{if } n = 0,1 \\ \binom{n-1}{k} * H_k * H_{n-1-k} & \text{if } n > 1 \end{cases}$$

where k=f(n) is a certain function of n that depends on the heap structure.

- A) Justify the correctness of this formula, and explain in words what k is.
- B) Prove that for all n we have $(n-1)/2$ ^{2} $k=f(n) \leq (2n-1)/3$.
- C) Show that at least one of k or n-1-k is a power of two minus one (2^h-1).
- D) Let $ñ=2^{[\lg n]}$ where [x] be the integer part of real number x. Show that

$$f(n) = \begin{cases} n - ñ/2 & \text{if } ñ \leq n < 3ñ/2 \\ ñ - 1 & \text{if } 3ñ/2 \leq n < 2ñ \end{cases}$$

[6pts]

5) Let N be an integer. Let [x] be the integer part of real number x.

- **A)** Describe an algorithm to compute the [lg N] in O(log N) time, and justify your running time.
- **B)** Describe an algorithm to compute the $[\sqrt{N}]$ in O(log N) time, and justify your running time.

[4pts]

6) Compare MergeSort with HeapSort in terms of

- **A)** Time used to sort n elements when implemented with arrays.
- **B)** Time used to sort n elements when implemented with linked lists.
- **C)** Space used to sort n elements when implemented with arrays.
- **D)** Space used to sort n elements when implemented with linked lists.

[8pts]

7) For each statement, say if it is *true* or *false*.

<u>Correct = +1 pt,</u> <u>Incorrect = -0.5 pt,</u> <u>No answer = 0 pt,</u> <u>Minimum Total= 0 pt.</u>

(a) The center of a tree (HW5) is always a single vertex or a single edge.

(b) The regular expressions **ab+a(b+a)*** and **a+a(a+b)*** generate the same strings.

(c) If g(n) **is** O(h(n)) then $2^{g(n)}$ **is** $O(2^{h(n)})$.

(d) Besides the case where n points are co-linear, the convex hull of n points in the plane contains as little as 3 line segments and as many as n line segments.

(e) We can easily modify any sorting algorithm to run in O(n) time in best case.

(f) The best cases of Binary Search is when the element we are seeking is absent.

(g) The 3-colorability of graphs is an undecidable problem.

(h) A tree is a forest.

[8pts]

8) Let A and B be nxn matrices where $n=2^k$ is a power of 2. Let M=AB be the matrix product of A and B to be defined recursively as

$$M=\begin{pmatrix} A_{11}B_{11}+A_{12}B_{21} & A_{11}B_{12}+A_{12}B_{22} \\ A_{21}B_{11}+A_{22}B_{21} & A_{12}B_{12}+A_{22}B_{22} \end{pmatrix}$$

where the original matrices are

$$A=\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}$$

$$B=\begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}$$

where each sub-matrix A_{ij}, B_{ij}, M_{ij} is an n/2xn/2 matrix (the base case is n=1). The above method uses <u>eight</u> multiplications of sub-matrices and <u>four</u> additions of sub-matrices.

- A) Write the recurrence equation that defines the running time T(n) needed to multiply two nxn matrices using the above method.

<u>You may use without proof the fact that adding two nxn matrices requires $\Theta(n^2)$ time. Operations on integers are at constant cost.</u>

- B) Solve this recurrence using the Master Method.
  <u>You do not need to prove BIG-O notation claims such as n^2 **is** $O(n^3)$…</u>

- C) Consider the following alternate way of computing the matrix product. One may prove that the matrix M defined below is the same as the matrix M defined above. Repeat A) and B) for this new method:

$$M=\begin{pmatrix} M_1+M_2 & M_3+M_5+M_6 \\ M_3+M_4-M_7 & M_3+M_4+M_5 \end{pmatrix}$$

where the <u>seven</u> n/2xn/2 sub-matrices $M_1,\ldots,M_7$ are computed as follows from the A_{ij}, B_{ij}.

$M_1=A_{12}B_{21}$

$M_2=A_{11}B_{11}$

$M_3=(A_{21}+A_{22}-A_{11})(B_{22}-B_{12}+B_{11})+M_2$

$M_4=(A_{11}-A_{21})(B_{22}-B_{12})$

$M_5=(A_{21}+A_{22})(B_{12}-B_{11})$

$M_6=(A_{12}-A_{21}+A_{11}-A_{22})B_{22}$

$M_7=A_{22}(B_{11}+B_{22}-B_{12}-B_{21})$