YORK UNIVERSITY **Dept. of Computer Science**
Faculty of Pure & Applied Science **May 14, 2001**

FINAL EXAMINATION
COSC3101: Design & Analysis of Algorithms

- Five problems, 180 minutes.
- **Crib Sheet:**
  - □ You may use a crib sheet with your name and student number on it as explained in the course.
  - □ Submit your crib sheet together with your exam booklet, when the exam is over.
- Do not spend too much time on a single problem. Read them all through first and solve them in the order that allows you to make the most progress.
- Do not use any electronic/mechanical computation devices.
- You will be graded not only on the correctness of your answer, but also on the clarity with which you express it. Be neat.
- Use the back of the pages for scratch work.
- This booklet consists of 9 pages, including this cover page.

Name:____

Student Number:________________________________

Problem	Worth	Mark
1	20	
2	15	
3	15	
4	25	
5	25	
TOTAL	100	

Problem 1. **[20%]**

Fill in the underlined blanks with the most appropriate and most simplified answer.

(a) If the average-case time complexity of an algorithm is $\Theta(n \; lg \; n)$, then its worst-case time complexity is ________________.

(b) $\sum_{i=1}^{n} (3 \; lg \; i + \frac{i}{n} + \frac{n \; lg \; n}{i}) = \Theta($___________________$)$.

(c) The solution of the recurrence $T(n) = 4T(\lfloor \frac{n}{4} + \sqrt{n} \rfloor) + 4n + \sqrt{n}$ is $T(n) = \Theta($_________________$)$.

(d) Name four greedy algorithms with their greedy choices learned in the course:
(i) __________________________________, greedy choice:__________________________________ ,
(ii) __________________________________, greedy choice:__________________________________ ,
(iii) __________________________________, greedy choice:__________________________________ ,
(iv) __________________________________, greedy choice:__________________________________ .

(e) Consider the integer sequence given by the recurrence $g_0 = 0$, $g_1 = 1$, and $g_n = 2g_{n-1} + 3g_{n-2}$, for all $n \geq 2$. By a fast technique learned in class, we can compute g_n, for an arbitrary given n, applying the ________________________ method, using $\Theta($__________$)$ arithmetic operations.

(f) Does the White-Path Theorem on Depth First Search (DFS) imply that if $u \rightarrow v$ is an edge of the given directed graph G, and DFS of G visits vertex u before v, then u will become DFS-parent of v? Answer: _____________.

(g) Does Kruskal's Minimum Spanning Tree algorithm guarantee to find an MST, even if some edges have negative weights? Answer: ____________ . What about Prim's MST algorithm? Answer: ______________.

(h) Consider a weighted complete graph on n vertices numbered 1 through n, where the weight of edge (i, j) is 2^{i+j}, for all distinct vertices i and j. Then, the weight of the Minimum Spanning Tree of this graph is exactly __________________.

(i) An efficient method to solve the single source **longest** paths problem on a weighted directed acyclic graph (DAG) is ____
____.

(j) Let f be a feasible flow in a capacitated flow network $G = (V, E)$ with source and terminal nodes s and t and edge-capacity function c. Then f is a max-flow from s to t in G if there is no directed path from s to t in the digraph __________________________________.

Problem 2. **[15%]**

The problem of computing the product of two polynomials is: given the coefficients of two degree $n-1$ polynomials $P(x)$ and $Q(x)$ as shown below, compute the coefficients of the product polynomial $P(x) * Q(x)$.

$$P(x) = \sum_{i=0}^{n-1} p_i x^i \;=\; p_0 + p_1 x + \cdots + p_{n-1} x^{n-1} \;,$$

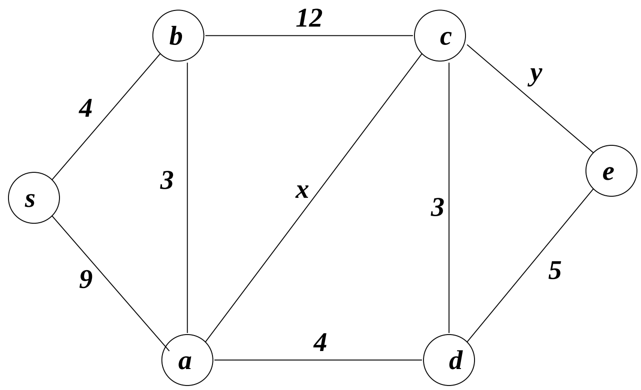
$$Q(x) = \sum_{i=0}^{n-1} q_i x^i \;=\; q_0 + q_1 x + \cdots + q_{n-1} x^{n-1} \;.$$

An elementary algorithm for computing the product takes quadratic time, i.e., $\Theta(n^2)$. There exists an asymptotically faster divide-and-conquer algorithm for this problem that works by solving *three* (rather than four) subproblems of size $n/2$.

Describe such a divide-and-conquer algorithm and by analyzing its time complexity show that it is subquadratic.

Problem 3. [15%]

Consider the following undirected graph with a length shown for each edge. The lengths x and y are unknown variables, and their values may be any integer greater than or equal to zero. For each of the questions below, fill in the appropriate answer, keeping in mind that the answer is with respect to the range of all possible values for x and y.



(a) [2%] The length of the shortest path from s to a is _______ and the length of the shortest path from s to b is _______, independent of the values of x and y.

(b) [3%] The length of the shortest path from s to c is minimum _______ and maximum _______.

(c) [3%] The length of the shortest path from s to d is minimum _______ and maximum _______.

(d) [3%] The length of the shortest path from s to e is minimum _______ and maximum _______.

(e) [4%] The length of the Minimum Spanning Tree is minimum _________ and maximum _________.

Problem 4. [25%]

Consider an array $A[1..n]$ of n arbitrary unsorted integers. We say A is α−modular if $|A[i] - A[i+1]| \le \alpha$, for all $i = 1..n-1$.

(a) [2%] What is the smallest α such that the following sequence is α-modular $(3, 2, 3, 5, 4, 2, 2, -1, -2, 0, 1, 1, 1, 2, 2)$?

(b) [8%] Show that if α is a constant, then an α-modular sequence can be sorted in linear time. [Hint: Use a bucketing technique. Explain the details.]

(c) [3%] The $(\alpha\ ,\ \beta)$-*LMS* problem, where $0 \le \beta \le \alpha$ is: given an α-modular sequence A of length n, compute a longest β-modular subsequence B of A. (Note: a subsequence is not necessarily contiguous.)

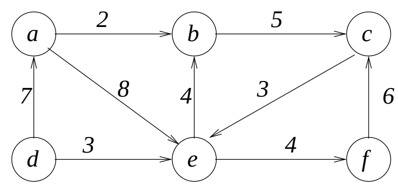
Briefly describe an $O(n)$ time algorithm to solve the $(\alpha\ ,\ 0)$-*LMS* problem, where α is some given constant integer ≥ 0.

(d) [12%] Describe an $O(n)$ time algorithm to solve the $(\alpha\ ,\ 1)$-*LMS* problem, where α is some given constant integer ≥ 1. (Slower algorithms may get partial credit.)

Problem 5. **[25%]**

The **single-source bottleneck path problem (SSBPP)** on weighted directed graphs is defined as follows: Given a directed graph $G = (V, E)$ with real valued edge costs, define the *bottleneck cost* of a path to be the maximum of the costs of the edges on the path. The optimum bottleneck path from a vertex s to a vertex t is a path from s to t in G with minimum bottleneck cost (i.e., the longest edge on the path is as short as possible). The problem is: given a source vertex $s \in V$ find the optimum bottleneck paths from s to each vertex of G.

(a) [2%] Consider the example digraph below.



(i) What is the bottleneck cost of the path (a, e, f)?

(ii) What is the optimum bottleneck path from vertex a to f?

(iii) What is the bottleneck cost of that path?

(b) [4%] Describe how to modify the initialization and the edge-relaxation (or labeling) operations of Ford's single-source shortest paths method (i.e., *Initialize-Single-Source(G,s)* and *Relax(u,v,w)*) for the solution of SSBPP.

(c) [3%] Would the existence of negative cost edges create any complication in part (b)? Explain.

(d) [5%] Describe an algorithm that solves SSBPP in $\Theta(V + E)$ time if G is a directed acyclic graph (DAG).

(e) [11%] Describe an efficient algorithm that solves SSBPP on a general digraph. Show an exact pseudo-code. Analyze its time complexity. [Hint: Modify Dijkstra's algorithm.]

- The End -