YORK UNIVERSITY

FACULTY OF PURE AND APPLIED SCIENCE

2001 FALL TERM EXAMINATION

Course Number: COSC3101
Title: Design and Analysis of Algorithms

Duration: 3 hours

Examination Aids: One sheet of paper measuring $8\frac{1}{2} \times 11$ inches, handwritten on both sides.

- There should be 9 pages in the exam, including this page.
- Write all answers on the examination paper. If your answer does not fit in the space provided, you can continue your answer on page 9 or on the back of a page, indicating clearly that you have done so.
- Write legibly.
- Throughout the exam, "running time" means "worst-case running time".

Name ______________________________

(Please underline your family name.)

Student Number ______________________________

Circle the name of your instructor

Eric Ruppert   Gordon Turpin

1.	______/5
2.	______/3
3.	______/3
4.	______/2
5.	______/3
6.	______/6
7.	______/9
8.	______/6
9.	______/12
10.	______/11
Total:	______/60

1. [5 marks] For each of the following questions, circle the number of the one correct answer.

(a) Let $f(n) = n\sqrt{n}\log n$. Which of the following is true?

1. $f(n) \in O(n^{1.5})$.
2. $f(n) \in \Omega(n^{1.5})$.
3. $f(n) \in \Theta(n^2)$.
4. $f(n) \in \Omega(n^2)$.
5. None of the above.

(b) Which of the following statements is true for all functions f and g?

1. If $f \in O(g)$ then $g \in \Theta(f)$.
2. Either $f \in O(g)$ or $g \in O(f)$.
3. If $f \in O(g)$ then $g \in \Omega(f + g)$.
4. If $f(n) \in O(g(n))$ then $f(2n) \in O(g(n))$.
5. None of the above.

(c) Let $P(0) = 2$ and $P(n) = (n/2)^3 + P(n-1)$, for $n > 0$. Which of the following is true?

1. $P(n) \in O(n^3)$.
2. $P(n) \in \Omega(n^4 \log n)$.
3. $P(n) \in \Omega(n^3 \log n)$.
4. $P(n) \in \Theta((n/2)^3)$.
5. None of the above.

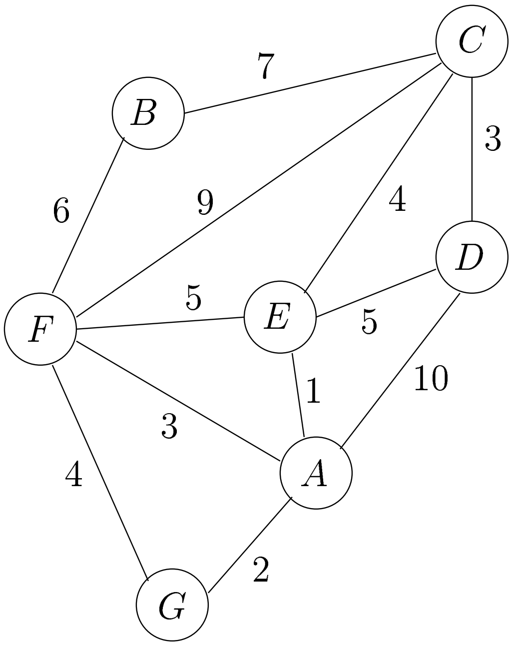
(d) Let $Q(0) = 1$ and $Q(n) = 6 \cdot Q(\lfloor 2n/5 \rfloor) + n^2$ for $n > 0$. Which of the following is true?

1. $Q(n) \in \Theta(n^{\log_{0.4} 6})$.
2. $Q(n) \in \Theta(n^{\log_6 2.5})$.
3. $Q(n) \in \Theta(n^{\log_{2.5} 6} \cdot \log n)$.
4. $Q(n) \in \Theta(n^2)$.
5. None of the above.

(e) Which of the following statements implies an $\Omega(\sqrt{n})$ lower bound on the worst-case running time of an algorithm A.

1. For all every large value of n, there is a $c > 0$ such that A takes at least $c\sqrt{n}$ steps on all inputs of size n.
2. There is a $c > 0$ such that for all large n, A takes at most $c\sqrt{n}$ steps on all inputs of size n.
3. For all $c > 0$, there is an n such that A takes at least $c\sqrt{n}$ steps on some input of size n.
4. There is some $c > 0$ and some n and some input of size n for which A takes at least $c\sqrt{n}$ steps.
5. There is a $c > 0$ such that for all large n, A takes at least $c\sqrt{n}$ steps on some input of size n.

2. [3 marks] Below is a map of seven towns connected by roads. The number beside each road is the cost of ploughing snow on that road. Choose a set of roads to plough so that there is a snow-free route from the fire station in town A to every other town. Your goal is to minimize the total amount of money spent on ploughing snow.



3. [3 marks] Suppose radix sort is executed on an array that initially contains the (decimal) numbers $7, 384, 95, 860, 982, 453, 560$. Show the contents of the array after each of the three passes performed by radix sort.

4. [2 marks] Is radix sort stable? Briefly justify your answer.

5. [3 marks] One can compute the length of the shortest path between every pair of vertices in a weighted, directed graph by running the Floyd-Warshall algorithm or by running Dijkstra's algorithm once for each source node in the graph. (Assume the priority queue in Dijkstra's algorithm is implemented using a heap.)

For large values of n, which of the two approaches above is preferable if the graph has n vertices and there are exactly $\lfloor \sqrt{n} \log n \rfloor$ edges leading out of each vertex? Briefly explain why.

6. [6 marks] Given three distinct elements A, B and C, you must find the median.

[4] **(a)** Draw a decision tree that solves the problem. (Your tree should have the smallest possible height, but you need not prove this.)

[2] **(b)** If all six orderings of the input elements ($A < B < C, A < C < B, B < C < A, B < A < C, C < A < B$ and $C < B < A$) are equally likely, what is the average-case number of comparisons performed by your algorithm? Show your work.

7. [9 marks] You are given a connected, undirected graph with n vertices and m edges. The edges are weighted.

[3] **(a)** Suppose the priority queue in Prim's algorithm is implemented using an array that contains, for each queue element, its key value. What would the running time be? Briefly explain why your answer is correct.

[6] **(b)** If you know that the weight of each edge is either 1, 2 or 3, how could you implement the priority queue in Prim's algorithm to make the algorithm run in $O(m)$ time? (Hint: you should be able to extract the minimum element and update keys in constant time.)

8. [6 marks] Andrew has C dollars to spend on Christmas presents for Jason. There are n different items for sale. Let c_i be the cost of item number i and let p_i be the probability that Jason will *not* like item i. Andrew wishes to buy a set of items that maximize the probability that Jason will like *at least one* of his presents. (Assume Jason's opinions of different items are independent of one another.) Define a dynamic programming array that can be used to compute this maximum probability and give a recurrence that can be used to fill in the array's entries. Include the base case(s). Explain why your recurrence is correct. (You do *not* have to explain how to output the actual set of items to buy.)

9. [12 marks] Leutonian coins have the following denominations: 19 klopecks, 29 klopecks, and 47 klopecks. Given n, you must determine whether or not there exists a collection of Leutonian coins whose total value is n klopecks. Let L be an array where

$$L[i] = \left\{ \begin{array}{ll} \text{true} & \text{if there is a set of coins with total value } i \\ \text{false} & \text{otherwise} \end{array} \right\}.$$

[3] **(a)** Give a recurrence that can be used to fill in L. Include the base case(s).

[2] **(b)** How much time would it take to fill in the entries of $L[1..n]$ using your recurrence?

[7] **(c)** Assuming $L[1..n]$ has already been computed, write an algorithm that outputs a set of coins with total value n. State pre- and post-conditions for your algorithm. If you use a loop, state its invariant. (Do not prove them.)

10. [11 marks] There are n gas stations along a highway. The distance, in kilometres, from station 1 to station i is d_i. Assume $0 = d_1 < d_2 < \cdots < d_n$ and that $d_{i+1} \leq d_i + 100$ for every i. You wish to drive from station 1 to station n, making as few stops at gas stations as possible. Your car holds enough gas to travel 100 km. The following greedy algorithm computes a list L of stations at which you should make stops.

Let L be a list containing 1
$i \leftarrow 1$
loop
    exit when $i = n$
    $i \leftarrow \max\{k : i < k \leq n \text{ and } d_k - d_i \leq 100\}$ % the next stop is the furthest one you can reach
    add i to end of L
end loop
output L

Let L_i be the value of L after i iterations.

[2] **(a)** What does it mean for an optimal solution L^* to extend L_i for this problem?

[9] **(b)** Prove that for every $i \geq 0$, there is an optimal solution that extends L_i.

(Nearly) blank page for extra workspace.