University of Toronto
CSC270
Final Exam

December 16, 1996
7:00pm
Duration: 3 hours

Closed book, no calculators

Name: ____________________

Student Number: ____________________

There are 8 questions and 0 marks total.

Answer all questions.

This exam paper should have 11 pages,
including this cover page.

1 – True or False	/ 0
2 – Short Answer	/ 0
3 – Number Representation	/ 0
4 – Graphs	/ 0
5 – More Graphs	/ 0
6 – Random Variables	/ 0
7 – Dynamic Programming	/ 0
8 – Simulation	/ 0
Total	/ 0

1 True or False — 0 points

Part A — 0 points When representing a number in binary in normalized form, the first significant bit need not be stored since it is always 0. ____________

Part B — 0 points The random seed determines whether a sequence of random numbers, generated by the linear congruential method, includes every possible number from zero to MAXINT. ____________

Part C — 0 points A C++ template class defined by `template<class T> class MyClass` can be used with any class in place of `T`. ____________

... continued

2 Short Answer — 0 points

Part A — 0 points In C++, how should you decide whether the body of a function should appear in the class definition?

Part B — 0 points Why is a table necessary for dynamic programming?

Part C — 0 points A *tree* of n vertices is an connected, acyclic, undirected graph containing $n-1$ edges. What property of the graph changes when an edge (any edge!) is removed from the tree? What property changes when an edge (any edge!) is added to the original tree?

Part D — 0 points Give two reasons that two's complement is better than one's complement for representing signed integers.

... continued

3 Number Representation — 0 points

When final marks are adjusted, they can be scaled and shifted. In other words, calculations are of the form $M' = aM + b$, where M is the original mark, M' is the new mark, and a and b are real numbers.

Part A — 0 points Using the equation $M' = aM + b$, what is the result M' using the following values: $M = 86.5$, $a = 1.25$, and $b = -4$.

Part B — 0 points Repeat the process of Part A using only two significant digits and keeping only two significant digits after each step. Use proper rounding and the normal rules of precedence. Show the step–by–step results of each of your calculations.

Part C — 0 points Based on your answers for Part A, what is the absolute error of your results in Part B? Show your work.

Part D — 0 points Based on your answers for Part A, what is the relative error of your results in Part B? Show your work.

... continued

4 Graphs — 0 points

Here's an adjacency list representation of a directed graph, where each node (on the left) lists the other nodes reachable from it by following an edge.

node	adjacent nodes
A	B, E, F
B	D
C	A
D	E, F
E	C
F	B, E

Part A — 0 points Draw the graph.

Part B — 0 points Starting from node A and processing edges in the order indicated by the adjacency lists, draw the depth first search tree. For example, at node A the edges are processed in order (A,B), then (A,E), then (A,F). Label each of the tree, forward, backward and cross edges.

Part C — 0 points Are there cycles in the graph? How do you know this by examining the depth first search tree?

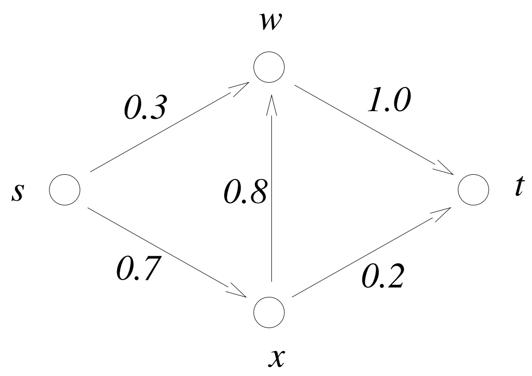
... continued

5 More Graphs — 0 points

On the Internet there are many different paths for a message to travel from a node s to a node t. It used to be that each message would decide on its path before leaving. This would cause congestion on certain paths. To avoid this, a message is now routed dynamically: Its path is determined as it travels through the network.

Let us represent a network as a directed, acyclic graph. For edge (u, v), let the weight $w(\ (u, v)\ )$ be the probability that a message leaving u will travel along the (u, v) edge. Since a message must leave u on *some* edge, the sum of the edge weights must be one.

In the graph below, edge weights are written beside the edges and node names are written beside the nodes. The probability that a message at x will travel to t is 0.2, since the weight $w(\ (x, t)\ )$ is 0.2.



Part A — 0 points There are three paths from s to t in the graph above. List the paths and the probability that each path will be taken. Be sure that the sum of your probabilities is one! Show your work.

... continued

Given a large graph representing a network as described above, we want an algorithm to find the *probability* of the most likely path between two given nodes: s and t. We *don't* want the most likely path, just its probability. One of the graph algorithms from class can be modified very slightly to accomplish this.

Part B — 0 points Which one of breadth–first search, depth–first search, Dijkstra's algorithm, and Floyd's algorithm should be modified to accomplish this? Ignore the issue of graph representation: You may assume that you get the graph in a representation appropriate for your chosen algorithm.

You will now describe the very slight modification that should be made to your chosen algorithm so that it computes the probability of the most likely path. This answer should be very short and clearly written. If it is long or unclear, you will not get full marks.

Part C — 0 points Describe how the data structures of the algorithm are changed.

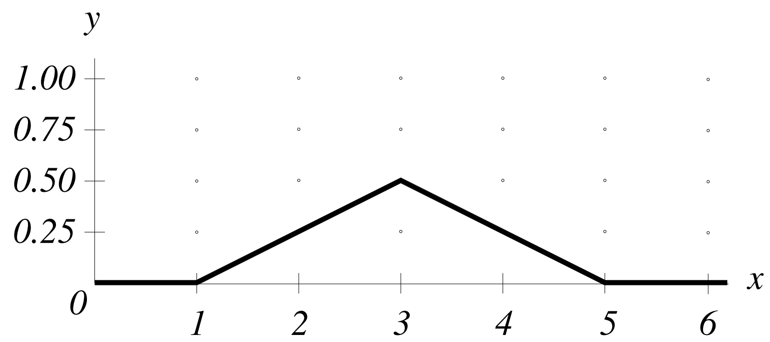
Part D — 0 points Describe how the code of the algorithm is changed.

... continued

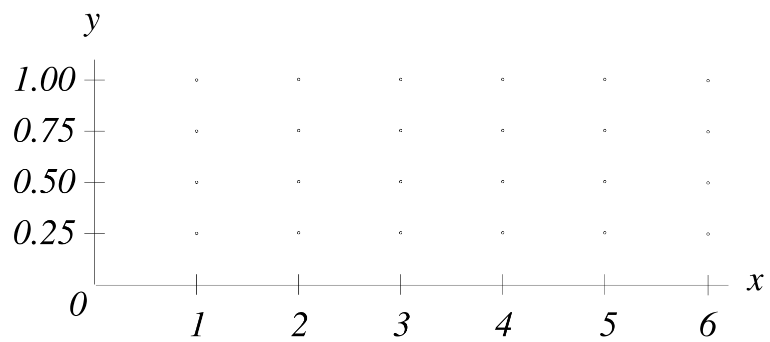
6 Random Variables — 0 points

Consider the following probability density function:

$$f_X(x) = \begin{cases} 0 & \text{if } x \leq 1 \\ (x-1)/4 & \text{if } 1 \leq x \leq 3 \\ (5-x)/4 & \text{if } 3 \leq x \leq 5 \\ 0 & \text{if } x \geq 5 \end{cases}$$



Part A — 0 points Draw the cumulative distribution function, $F_X(x)$, on the graph below.



Part B — 0 points What is the inverse of the cumulative distribution function, $F_X^{-1}(y)$. Write it in four cases (like $f_X(x)$ is written above). Show your work.

... continued

7 Dynamic Programming — 0 points

The *string cutting* problem is to break a character string into several substrings. However, the only allowable operation is to cut a string into two substrings. A single cut takes n time units for a string of length n, since it makes a copy of the string before cutting it. The running time does *not* depend upon the position of the cut.

Consider a string *abcdefghij* that we want to cut after c and after e. Here's what we can do:

Strings	Action	Cost
abcdefghij		
	cut *abcdefghij* after c	10
abc *defghij*		
	cut *defghij* after e	7
abc *de* *fghij*		

The total cost is 17.

Part A — 0 points What is the cost if we first cut after e, then after c? Show your work with a table similar to that above.

... continued

Suppose we want to cut a string into six fragments. Listed in order, these fragments are $s_1\ s_2\ s_3\ s_4\ s_5\ s_6$. The fragments have the following lengths:

Fragment	s_1	s_2	s_3	s_4	s_5	s_6
Length	1	1	3	1	2	2

In other words, we're taking a string like *abcdefghij* and cutting it into strings *a*, *b*, *cde*, *f*, *gh*, and *ij*.

You will use dynamic programming to find the sequence of cuts that has minimum total cost. In doing so, you will fill in the two tables below with your intermediate results.

In the left table, the element in row i, column j contains the minimum total cost of cutting the subsequence $s_j, s_{j+1}, \ldots, s_{j+i}$. Note that row numbers increase upward.

In the right table, the element in row i, column j contains the index of the fragment after which the first cut should be made. For example, if the first cut of $s_3\ s_4\ s_5$ should be made after s_4, then the row 2, column 3 entry in the right table will contain a "4".

Part B — 0 points Use dynamic programming to fill in the two tables.

5					
4					
3					
2					
1					
	1	*2*	*3*	*4*	*5*

minimum costs

5					
4					
3					
2					
1					
	1	*2*	*3*	*4*	*5*

cutting positions

Part C — 0 points From your tables above, what is the minimum cost cutting sequence?

... continued

8 Simulation — 0 points

... End Of Final Exam