University of Toronto
Scarborough Campus
December 16, 2009

CSC C73 Final Examination
Instructor: Vassos Hadzilacos

Aids allowed: One 8.5×11 handwritten, non-photocopied 'cheat sheet'

Duration: Three hours

- There should be 10 pages in this exam booklet, including this cover page.
- Answer all questions.
- Put all answers in this booklet, in the spaces provided.
- For rough work, use the backs of the pages; ***these will not be marked***.
- Good luck!

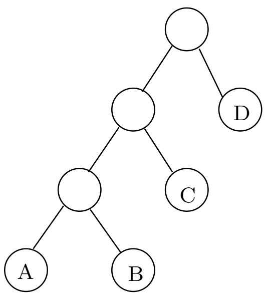
Family Name ______________________ Given Name ____________________

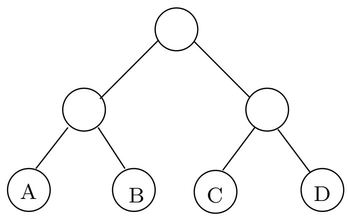
Student Number ____

Problem	Marks Rec'ved	Marks Worth
1.		20
2.		10
3.		20
4.		10
5.		20
6.		20
7.		20
8.		30
TOTAL		150

QUESTION 1. (20 marks)

a. (10 marks) Fill the frequency table below with a frequency for each of the symbols A, B, C, and D so that, for these frequencies, different executions of Huffman's algorithm (corresponding to different ways of resolving ties) could produce *both* of the following two trees.



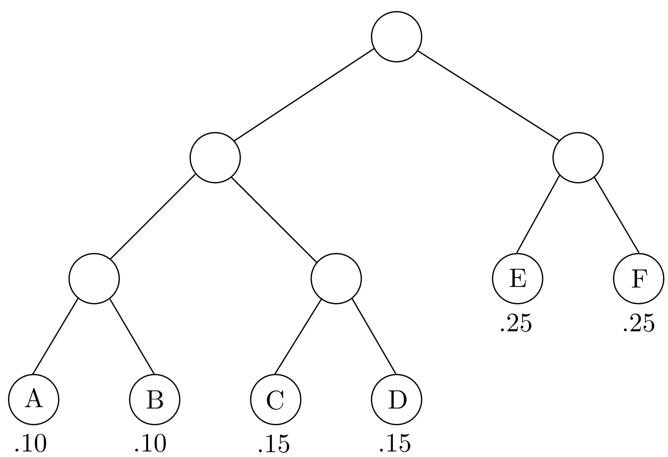


ANSWER:

Symbol	Frequency
A	
B	
C	
D	

b. (10 marks) Below is a full binary tree that represents an encoding of the symbols A, B, C, D, E, and F (shown as leaves of the tree), where each symbol appears with the frequency indicated below the leaf that corresponds to the symbol.

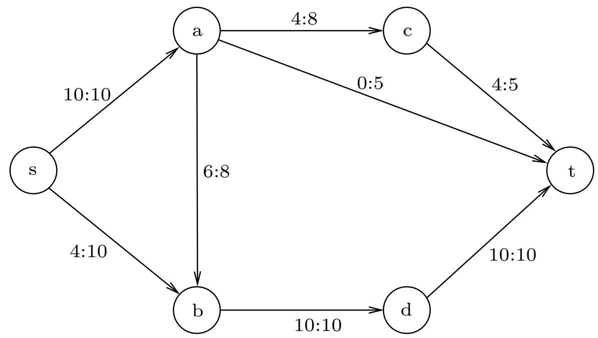
Is the encoding represented by this tree optimal? That is, does it minimise the number of bits required to encode a text over the six symbols A-F, where the symbols appear with the indicated frequencies? Justify your answer.



ANSWER:

QUESTION 2. (10 marks)

Shown below if a flow graph, and a flow through its edges. The label "$f : c$" on an edge indicates that the capacity of the edge is c and the flow through the edge is f.



a. (7 marks) Is the flow shown a maximum flow? Justify your answer.

ANSWER:

b. (3 marks) Find a minimum cut of the above flow graph.

ANSWER:

QUESTION 3. (20 marks)

Dijkstra's algorithm does not always work correctly if edges can have negative weights. To rectify this problem Professor N. O'Bright suggests the following: Let w be the minimum (negative) weight of any edge; add $|w|$ to every edge's weight, thereby ensuring that each edge has a non-negative weight; then use Dijkstra's algorithm to find a shortest path from the start node s to each vertex in this modified graph.

Does N. O'Bright's idea work? That is, is a shortest path from s to each vertex v in the modified graph also a shortest path from s to v in the original graph? Justify your answer.

ANSWER:

QUESTION 4. (10 marks)

Suppose you have access to a software package that performs efficiently basic linear algebra operations including matrix inversion and multiplication of a matrix by a vector.

How would you use this package to determine the coefficients of the degree 4 polynomial $p(x)$, for which $p(1) = 2$, $p(-1) = -3$, $p(2) = -2$, $p(-2) = 1$, and $p(3) = 5$? (You do not actually have to find the coefficients here; you only need to describe the steps by which they can be obtained.)

ANSWER:

QUESTION 5. (20 marks)

Let $A[1..n]$ be a ***sorted*** array of ***distinct*** integers. Describe a divide-and-conquer algorithm that determines whether there exists some i, $1 \leq i \leq n$, such that $A[i] = i$. Your algorithm should run in $O(\log n)$ time. Briefly explain why your algorithm is correct, and why it achieves the stated running time.

ANSWER:

QUESTION 6. (20 marks)

We are given a positive integer k, and an array $A[1..n]$ that contains n distinct positive integers. We wish to determine whether there is a subset of the integers contained in A whose sum is equal to k. Give a dynamic programming algorithm that does this: the algorithm returns 1 if such a subset exists and 0 otherwise. Your algorithm should run in $O(nk)$ time. Briefly explain why your algorithm works.

ANSWER:

QUESTION 7. (20 marks)

There are m mobile phone users $u_1, \ldots, u_m$ and n base stations $s_1, \ldots, s_n$. Based on the users' present location and the location of the base stations, for each user u_i, $1 \leq i \leq n$, we are given a set $X_i \subseteq \{s_1, \ldots, s_n\}$ of the base stations to which u_i can (presently) connect. Each base station s_j has a capacity ℓ_j, that is the maximum number of users who can connect to it. We want to determine whether it is possible to connect all users to base stations without exceeding the capacity of any base station.

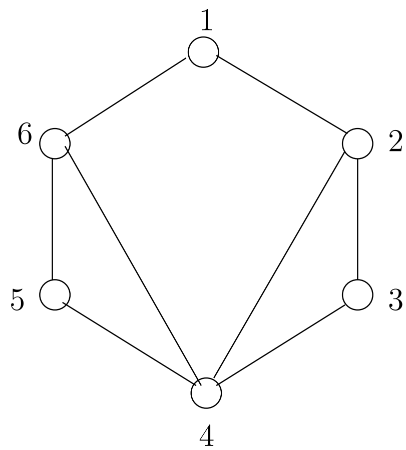
Give a polynomial-time algorithm that solves this problem. Explain why your algorithm is correct, and analyse its running time.

ANSWER:

QUESTION 8. (30 marks)

Let $G = (V, E)$ be an undirected graph. A ***matching*** of G is a subset of the edges $M \subseteq E$ such that every vertex in V is an endpoint of at most one edge in M. A matching M is ***maximal*** if no proper superset of M is also a matching of G; it is ***maximum*** if there is no matching of G with more edges than M.

a. (5 marks) Identify a maximal matching that is not a maximum matching of the graph below.



ANSWER:

b. (10 marks) Let M be a maximal matching of a graph $G = (V, E)$, and let V_M be the subset of nodes that are endpoints of the edges in M. Prove that (i) V_M is a vertex cover of G; and (ii) for any vertex cover C of G, $|C| \geq |V_M|/2$. (Recall that a vertex cover of a graph is a subset V' of the nodes such that every edge has at least one endpoint in V'.)

ANSWER:

[Continued on the next page]

c. (15 marks) Give an approximation algorithm that takes as input a graph G in adjacency list form, and returns a vertex cover of G whose size is at most twice the size of a minimum vertex cover of G. Your algorithm should run in $O(n + m)$ time, where n is the number of vertices and m is the number of edges of G. Explain why your algorithm is correct.

Hint. Use the property of maximal matchings that you proved in part (b). Note that the bound on the size of V_M that you proved there holds for *any* vertex cover C — and so, in particular, it holds when C is a minimum vertex cover.

ANSWER:

THE END