CS 3AC3 Midterm Test

Friday, March 1, 2024, 7:30pm-9:30pm, Total: 40 pts

Virtual Take Home Midterm Test

- THIS EXAMINATION PAPER HAS **3** PAGES AND **8** QUESTIONS.
- The exam starts at 7:30pm and ends at 8:50pm, i.e. 80 minutes (for students without extra time permissions). This includes the exam time (60 minutes) plus extra time for technology (20 minutes).
- Please submit the solutions via Avenue using the same procedure as for the assignments.
- Any question about the test questions, ask by sending an e-mail to Ryszard Janicki (janicki@mcmaster.ca), while any question about Avenue ask by sending and e-mail to Holly Koponen (koponeh@mcmaster.ca).

1.[4] a.[2] Using only definition of O(f(n)) prove that the following statement is true:

$$2n^2/(\log n + 1) = O(n^2)$$

b.[2] Using only definition of O(f(n)) prove that the following statement is false:

$$2n^2 + 1 = O(n)$$

2.[4] Give tight big-oh and big-omega bounds on T(n) defined by the following recurrences.. Assume T(1)=1.

a.[2] $T(n)=2T(n/3)+4 \log n$

b.[2] $T(n)=2T(n/2)+12n^2$

3.[4] Is the following statement true?

In every instance of the Stable Matching Problem, there is a stable matching containing a pair (m,w) such that m is ranked first on the preference list of w and w is rank first on the preference list of m.

If ‘true’, give a short explanation. If ‘false’, give a counterexample.

4.[6] Consider the following *Interval Partitioning* problem for 6 jobs/lectures:

Job #	Start time	Finish time
1	1	2
2	1	3
3	8	10
4	7	11
5	3	8
6	4	6

All jobs/lecture must be done between times 1:00 and 11:00. Find minimum number of machines/classrooms. Use greedy algorithm. Give all steps. The problem is simple enough to be solved without using any formal technique but such solution will not be accepted.

5[4] Let *a*, *b*, *c*, *d* be four characters with probabilities 0.55, 0.10, 0.25, 0.10, respectively. Construct an optimal Huffman code and compute the average code length.

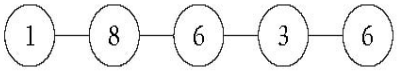
6.[6] Let $G = (V, E)$ be an undirected graph with n nodes.

A subset of the nodes is called an *independent set* if no two of them are joined by an edge.

Finding large independent sets is difficult in general; but here we'll see that it can be done efficiently if the graph is "simple" enough.

Call a graph $G = (V, E)$ a *path* if its nodes can be written as $v_1, v_2, \ldots, v_n$, with an edge between v_i and v_j if and only if the numbers i and j differ by exactly 1. With each node v_i, we associate a positive integer *weight* w_i.

Consider, for example, the five-node path drawn below.



The *weights* are the numbers drawn inside the nodes. The maximum weight of an independent set is 14.

Give an algorithm that takes an n-node path G with weights and returns an independent set of maximum total weight. The running time should be polynomial in n, independent of the values of the weights.

7.[6] Show the actions of the divide-and-conquer integer multiplication algorithm (with Karatsuba trick) when multiplying 1011 by 1101.

8.[6] The number of combinations of m things out of n, C(n,m) is given by the following formula:

$$C(n,m) = \begin{cases} 1 & \text{if } m=0 \text{ or } m=n \\ C(n-1,m)+C(n-1,m-1) & \text{if } 0 < m < n \end{cases}$$

Give a dynamic programming algorithm to compute C(n,m) and illustrate it by providing a filled table that is needed to compute C(5,3).

Hint. Pascal triangle as 2-dimensional table