University of Toronto
Scarborough Campus
October 23, 2020

CSC C73 Midterm Test
Instructor: Vassos Hadzilacos

Aids allowed: One 8.5×11 'cheat sheet' (may be written on both sides)

Duration: Two hours

READ THE INSTRUCTIONS CAREFULLY

- There should be 6 pages in this exam booklet, including this cover page.
- Answer all questions.
- If possible, print the exam booklet, and put all answers in this booklet, in the spaces provided. If you do not have access to a printer, ***write the answer to each question on a different sheet of paper***.
- At the end of the exam period scan or take a photograph of your answers, and upload them on Crowdmark in PDF, JPG, or PGN form. Make sure that you upload the correct file for each answer, and that the images you upload are oriented correctly.
- In your answers you may use any result discussed in this course or its prerequisites merely by stating it.
- Good luck!

Problem	Marks Rec'ved	Marks Worth
1.		30
2.		10
3.		20
4.		30
5.		30
TOTAL		120

QUESTION 1. (30 marks)

For each part below, circle ***all and only*** the choices (i)–(iv) that complete the sentence to a true statement. If no choice completes the sentence to a true statement, do not circle any of them. Do not justify your answers.

a. Dijkstra's algorithm
(i) solves the single-source shortest path problem if all edge weights are non-negative.
(ii) may run into an infinite loop if some edge weights are negative.
(iii) may return a path of smaller weight than the minimum-weight path if some edge weights are negative.
(iv) may return a path of greater weight than the minimum-weight path if some edge weights are negative.

b. Dijkstra's algorithm implemented using heaps and applied to a graph with n nodes and m edges
(i) runs in $O(n+m)$ time.
(ii) runs in $O(n \log m)$ time.
(iii) runs in $O(n \log n)$ time if every node has a constant number of outgoing edges.
(iv) runs in $O(n^2)$ time, if every node is connected to every other node.

c. Huffman's algorithm applied to an alphabet of $n \geq 2$ symbols
(i) can produce a codeword of length $\Omega(n)$.
(ii) will necessarily produce a codeword of length $O(\log n)$.
(iii) will necessarily produce two codewords of the same length.
(iv) runs in $O(n)$ time.

d. Karatsuba's algorithm
(i) solves the integer multiplication problem.
(ii) runs in $O(n \log n)$ time, where n is the maximum number of bits of the numbers it multiplies.
(iii) is an example of a greedy algorithm.
(iv) is an example of a divide-and-conquer algorithm.

e. If n is a power of 2 and ω is an n-th root of 1, then
(i) 2ω is also an n-th root of 1.
(ii) $\omega/2$ is also an n-th root of 1.
(iii) ω^2 is also an n-th root of 1.
(iv) $-\omega$ is also an n-th root of 1.

f. The Fast Fourier Transform on n-points
(i) runs in $O(n \log n)$ time.
(ii) can be proved to be correct by using the promising set approach.
(iii) can be used to multiply an arbitrary $n \times n$ matrix by the vector consisting of the n-th roots of 1.
(iv) can be used to evaluate any polynomial of degree $n-1$ at the n-th roots of 1.

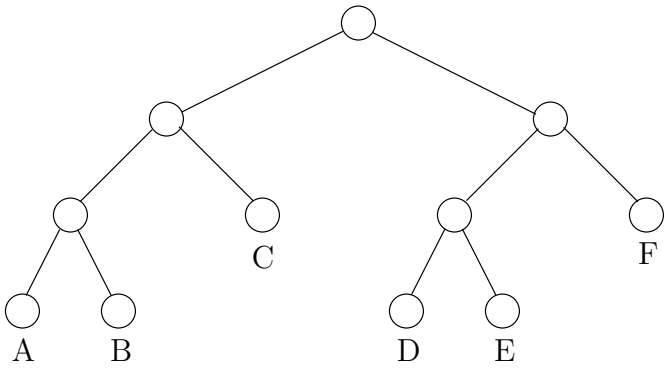
g. If $f(n) = 8f(n/2) + n^3$ and $g(n) = 27g(n/3) + n^2$ then
(i) $f(n) = O\big(g(n)\big)$.
(ii) $g(n) = O\big(f(n)\big)$.
(iii) $f(n) = \Theta\big(g(n)\big)$.
(iv) $g(n) = \Theta\big(f(n)\big)$.

h. The randomized algorithm for order statistics
(i) runs in $O(n)$ expected time.
(ii) runs in $O(n \log n)$ time in the worst case.
(iii) reduces the input size to at least three quarters in every recursive call.
(iv) reduces the input size to at most three quarters in every recursive call.

QUESTION 2. (10 marks)

Shown below is a table giving the frequencies of the six symbols of an alphabet, and a binary tree representing a prefix code for these symbols. Does the tree represent an optimal code for these symbols and frequencies? Justify your answer.

Symbol	Frequency
A	4/32
B	3/32
C	9/32
D	4/32
E	3/32
F	9/32



ANSWER:

QUESTION 3. (20 marks)

We are given an array $A[1..n]$ of integers in arbitrary order, representing the incomes of individuals in a certain population.

Describe an algorithm that computes the average income of this population, ***excluding*** the richest 10% and the poorest 10% individuals in the population. Your algorithm should run in $O(n)$ time in the worst case. For simplicity, assume that incomes in $A[1..n]$ are distinct, and that n is a multiple of 10.

Describe your algorithm in ***clear and concise point-form English*** or in ***pseudocode***. Do not justify the correctness of your algorithm. Briefly justify its running time.

ANSWER:

QUESTION 4. (30 marks)

We are given a directed graph $G = (V, E)$, nodes $s, t \in V$, and an edge weight function $\mathbf{wt}$ such that $\mathbf{wt}(e) \geq 0$ for every $e \in E$. In addition, we are given a set E' of ***potential edges***, i.e., pairs of nodes that are not edges in E; and for each $e' \in E'$ we are given a non-negative weight $\mathbf{wt}(e')$.

We want to choose a ***single*** potential edge in E' that, if added to the graph, reduces as much as possible the weight of a shortest $s \to t$ path. (If there is no potential edge whose addition to G reduces the weight of the shortest $s \to t$ path, we can choose any potential edge.)

Describe an algorithm that solves this problem in $O\big(k + (m+n)\log n\big)$ time, where $n = |V|$, $m = |E|$, and $k = |E'|$. The graph is given in adjacency list form, and the set of potential edges (and their associated weights) is given as a list. Your algorithm ***must*** use an algorithm we discussed in this course.

Describe your algorithm in ***clear and concise point-form English***. Do not write pseudocode. Do not justify the correctness of your algorithm. Explain why your algorithm meets the required running time.

ANSWER:

QUESTION 5. (30 marks)

We are given an array $P[1..n]$, where $P[i]$ contains the price of a single stock in the ith day of a period of n days. We want to find a pair of indices (i, j) such that $1 \leq i \leq j \leq n$ and $P[j] - P[i]$ is as large as possible. (This is a pair of days such that if we bought the stock on day i and sold it on day $j \geq i$ we would maximize our profit — or minimize our loss. Note the requirement $i \leq j$: We cannot sell the stock before we buy it!)

For example, if $P = (5, 11, 2, 1, 7, 9, 0, 7)$ the unique (in this case) answer is $(4, 6)$ since $P[6] - P[4] \geq P[\ell] - P[k]$ for all k, ℓ such that $1 \leq k \leq \ell \leq 8$.

It is obvious how to find (i, j) in $O(n^2)$ time. In ***clear and concise point-form English*** or in ***pseudocode*** describe a ***divide-and-conquer algorithm*** that solves this problem in $O(n \log n)$ time. Briefly but convincingly justify the correctness of your algorithm, and analyze its running time.

Note: There is a divide-and-conquer algorithm that solves this problem in $O(n)$ time, and such a solution will receive a small amount of extra credit. If you don't see how to do this right away, don't spend time on it until after you have finished the rest of the exam.

ANSWER:

THE END