CMPT 307

Summer 2020

Final Exam

This is an open-book exam. You may also use your notes. You may not use the web or net except to (1) access an electronic copy of the textbook, if this is how you purchased the book, or (2) use a website as a document preparation tool (e.g. word processor), and (3) submit your answers.

The exam is set for 3 hours, from 3:30 to 6:30. You have a 15-minute grace period after the examination in which to prepare your PDF file and verification photo and submit them to CourSys under the Final Exam submission section. This means you have until 6:45 to submit the exam. Exams submitted after 6:45 will be penalized at a rate of 3 percentage points per minute until 7:00. Exams will not be accepted after 7:00.

Part of what is being examined is your familiarity with the concepts and the language used to describe them. Therefore, we will not take questions on the questions of the exam. Answer them to the best of your ability.

By submitting a response to this exam you are certifying that you have abided by all exam regulations and instructions. This includes that you are submitting your own work that you did not obtain from someone or someplace else. Keep it honest.

1. (4 points) What is the dual of a linear program? How does it relate to the primal? The more complete your answer, the better your marks.
2. (4 points) Fully describe the adjacency-list representation of a graph (not a digraph). How do you attach an attribute, such as a weight, to an edge in this representation?
3. (4 points) Suppose I have the results of running an all-pairs shortest path (APSP) problem on a graph. I want to find the shortest path from some vertex u to another vertex v. How can I read this from the results of the APSP? Give pseudocode.
4. (4 points) What properties must a problem have in order to have a dynamic programming solution? Explain these properties.
5. (4 points) Suppose problem Q is in P, and we can reduce Q to R in polynomial time. What does this say about R? In particular, is R also in P? And also, is R in NP?

6. (80 points; 20 each) Answer four of the following six questions. If you answer more than four, we will mark all questions and award you the four LOWEST marks you achieve, so it is not to your advantage to answer more than four. Clearly indicate which four you answer.
   a. An *inversion* in an array A[1..n] is a pair of indices (i, j) such that i < j and A[i] > A[j]. The number of inversions in an n-element array is between 0 (if the array is in sorted order) and n choose 2 (if the array is sorted backward). Describe and analyze an algorithm to count the number of inversions in an n-element array in O(nlogn) time.
   b. Prove that any connected acyclic graph with n ≥ 2 vertices has at least two vertices with degree 1. The degree of a vertex is the number of edges it has.
   c. Suppose you are given an array A[1 .. n] of integers, each of which may be positive, negative, or zero. A contiguous subarray A[i .. j] is called a *positive interval* if the sum of its entries is greater than zero. Describe and analyze an algorithm to compute the minimum number of positive intervals that cover every positive entry in A.
   d. A *cyclic rotation* of a string *S* is a string βα where *S* = αβ. (α and β are substrings and may be empty). For example, a cyclic rotation of **cackle** is **ckleca.** Its other cyclic rotations are **cackle, acklec, klecac, lecack,** and **ecackl**. Given a pattern P of m characters, and a text T of n characters, where m is small compared to n, describe and analyze an efficient algorithm to determine if T contains a cyclic rotation of P as a substring.
   e. Let a *vertex-edge flow network* be a flow network (with source vertex s and target vertex t) where both the edges and the vertices have constraints. The edge constraints are as in a normal flow network, and the vertex constraints operate to constrain the total positive flow going into a vertex. For instance, a vertex with a constraint of 3 would only allow 3 flow to pass through it. Describe and analyze an efficient algorithm to solve the maximum-flow problem in a vertex-edge flow network.
   f. Suppose we are dealing with d-dimensional boxes where box X has side lengths of (x_1, x_2, … , x_d) and box Y has side lengths of (y_1, y_2, … , y_d). We say that X *fits within* Y if there is a permutation π on {1, 2, … , d} such that $x_{\pi(1)} < y_1$, $x_{\pi(2)} < y_2$, … , $x_{\pi(d)} < y_d$. Do both (i) and (ii):
      i. Describe and analyze an efficient method to determine if one box fits within another.
      ii. Suppose you are given a set of n d-dimensional boxes {B_1, B_2, … B_n}. Describe and analyze an efficient algorithm to determine the longest sequence <$B_{i(1)}$, $B_{i(2)}$, … , $B_{i(k)}$> of boxes such that $B_{i(j)}$ fits within $B_{i(j+1)}$ for j = 1, 2, …, k-1. Express the running time in terms of n and d.