CMPT 307
Fall 2020
T. Shermer

Final

100 points. You are allowed to look at your notes, books, and the lecture slides. You are not allowed to search or roam the web. This exam is scheduled for 2 hours and 50 minutes. You then have 20 minutes to scan (if necessary) and upload your answers to CourSys as a PDF.

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 under your own name, and you did not obtain any parts of it from someone or someplace else. Keep it honest!

Put answers to separate problems on separate pages. Put your pages in the same order as the questions in your PDF file. Answer the subparts of question 4 on separate pages, but question 1 can go all on one page.

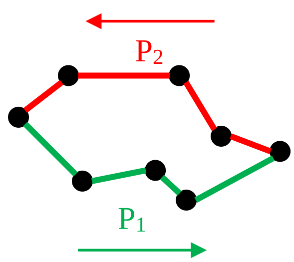
1. (15 points; 3 each)
   - a. What does $M \subseteq \Sigma^*$ mean? What does this define M as?
   - b. What is an *augmenting path*?
   - c. What is an *objective function*?
   - d. Explain the concept of *memoization*.
   - e. Define a *topological order* on a directed graph.

2. (5 points) Prove that P is a subset of NP.

3. (8 points) Explain why Johnson's algorithm for the All-Pairs Shortest Path problem uses vertex weights in order to reweight edges, rather than the simpler procedure of finding the minimum edge weight w_{min} (often a negative number) and subtracting that from every edge weight (thus giving all positive edge weights).

4. (72 points; 18 each) Answer four of the following seven problems. If you answer more than four, we will mark all of them and you will get the **lowest** four marks. So it is to your advantage to answer four only.

a. A *Monotone Euclidean Circuit* (MEC) for a point set P in two dimensions is an ordering of P that can be broken into P_1 followed by P_2 where P_1 and P_2 share endpoints and P_1 is ordered left-to-right and P_2 is ordered right-to-left (see figure). The *length* of a MEC is the sum of the lengths between consecutive points of P. Describe an $O(n^2)$ algorithm for finding a minimum-length MEC. Assume that P has no two points with the same x-coordinate. (Hint: scan from left to right, maintaining optimal possibilities for the two parts of the tour.)



b. Suppose you are given a set of tasks S={a_1, a_2, … a_n} with each task a_I having duration t_i. You have one processor which can run one task at a time, and each task must run to completion once it has started. Given a schedule (ordering of the tasks), the completion time c_i is t_i time units after the start of a_i. Give an algorithm to find a schedule that minimizes the average of the completion times $\frac{1}{n}\sum_{i=1}^{n} c_i$. **Prove** that the algorithm is correct (proof is worth 12 of the 18 points for this problem), and state its running time.

c. Suppose that the transitive closure of a directed acyclic graph G = (V, E) can be computed in O(f(m, n)) time, where n = |V| and m = |E|, and f(m, n) always increases when m increases and when n increases. Show that the time to compute the transitive closure of a general directed graph G = (V, E) is $O(f(m, n) + m^* + n)$, where m^* is the number of edges in the transitive closure.

d. A *spanning path network* in a graph G = (V, E) is a set of vertex-disjoint paths in G such that every vertex is in exactly one path. Paths may start and end anywhere, and they may be any length, including 0. A *minimum spanning path network* of G is a spanning path network of G containing the fewest possible paths.

   Give an efficient algorithm to find a minimum spanning path network of a directed acyclic graph G = (V, E). (Hint: assuming that V = {1, 2, 3, … n}, construct a graph G' on vertices $\{s, t\} \cup \{x_1, x_2, \ldots x_n\} \cup \{y_1, y_2, \ldots y_n\}$ and run a maximum flow algorithm on it.)

e. Strassen's Algorithm runs in time $O(n^{\log 7})$. Suppose you are given an integer k, a $kn \times n$ matrix A, and an $n \times kn$ matrix B. Using Strassen's algorithm as a subroutine, how can you efficiently multiply A*B? Also, how can you efficiently multiply B*A? Give algorithms and time complexity analysis.

f. **Clearly** describe why, in INITIALIZE-SIMPLEX, after the first pivot, the auxilliary program has a feasible basic solution.

g. Suppose we are matching a string pattern P against a text T. The text T uses the alphabet Σ and the pattern P uses the alphabet Σ ∪ {*} where * is a wildcard character that matches any string, including the empty string. For example,

   P = ad*dbc*ba

   matches the text

   T = dbadbbadbcbaaccadd

   because the first wildcard matches "bba" and the second wildcard matches the empty string. The wildcard character may occur an arbitrary number of times in the pattern. The pattern also matches

   T = bdcaadcbdacdbcbcbcdabaaabcdcb

   Give a polynomial-time algorithm to tell, given P of length m and T of length n, whether P matches the text T. Analyze the time complexity of your algorithm.