Qualifying Examination
Theoretical Computer Science
Monday, October 7, 2019
Part I: Algorithms

Instructions:

(a) This is a closed book exam.

(b) The exam has four problems worth 25 points each. Read all the problems carefully to see the order in which you want to tackle them. You have all day (9am–5pm) to solve the problems.

(c) Write clearly and concisely. You may appeal to some standard algorithms/facts from textbooks unless the problem explicitly asks for a proof of that fact or the details of that algorithm.

(d) If you cannot solve a problem, to get partial credit write down your main idea/approach in a clear and concise way. For example you can obtain a solution assuming a clearly stated lemma that you believe should be true but cannot prove during the exam. However, please do not write down a laundry list of half-baked ideas just to get partial credit.

(e) In problems with multiple parts, if you cannot solve a part, you can still solve subsequent parts (assuming that claims from earlier parts are true) and get partial credit.

May the force be with you.

Problem 1: Let $G = (V, E)$ be directed graph. Describe a polynomial time algorithm to find the fewest number of edges from $(V \times V) \setminus E$ that need to be added to G to make it strongly connected.

Hint: How would you solve the problem if G is a DAG?

Note: Partial credit will be given for an algorithm with running time $n^{O(k)}$ to decide if at most k edges are needed.

Problem 2: Describe and analyze an algorithm to determine whether one string X occurs as two disjoint subsequences of another string Y. For example, the string `PPAP` appears as two disjoint subsequences in the string `PENPINEAPPLEAPPLEPEN`; for example, `PenPineAppleaPplepen` and `penpineaPPleApplePen`.

Problem 3: In the game Ballmania, you are given a holy urn with n balls $b_1, \ldots, b_n$, labeled by $1, \ldots, n$, respectively. Initially, the weights of the balls are all one (i.e., $w_i = 1$, for all i). In each round, you pick a ball, where the probability of picking the ith ball b_i is $w_i / \sum_j w_j$. If you pick the ball b_i, then its weight becomes $w_i \leftarrow w_i + 1$. The game then continues to the next round, with the updated weights.

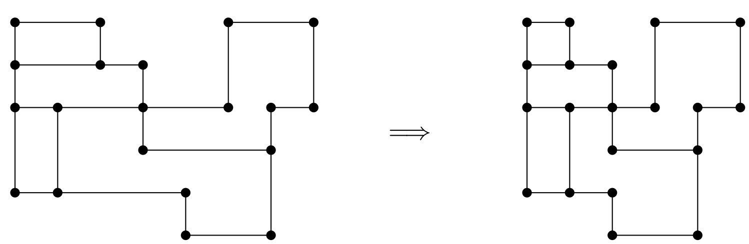
Provide upper and lower bounds, as tight as possible (ignoring constant factors), on the number of rounds, till all the n balls in the urn are encountered, and this happens with probability at least half.

(*Hint:* What is the probability of a ball that was not picked yet, to be picked in the ith round?)

Problem 4: We are given an embedding Γ of a planar graph of G with n vertices, where every edge is drawn as a horizontal or vertical line segment (without edge crossings), and every vertex is placed at a grid point (i.e., has integer x- and y-coordinates). We want a redrawing Γ' of G such that every edge is still drawn as a horizontal or vertical line segment (without edge crossings), the structure of the faces is unchanged, and the y-coordinates of all the vertices are unchanged, but the x-coordinates may change (though they must still be integers). Call such a redrawing Γ' a *horizontal compaction* of Γ.

(a) Describe an efficient algorithm to compute a horizontal compaction that minimizes the width, i.e., the difference between the maximum and minimum x-coordinates. You should reduce the problem to a path problem in a DAG.

(b) Describe an efficient algorithm to compute a horizontal compaction that minimizes the sum of the lengths of the horizontal edges. You should reduce the problem to minimum-cost flow or linear programming.



(a horizontal compaction of min width)