Assignment 2

COMP 5703 - Fall 2015

November 9, 2015

1 Instructions

This is due at the start of the class on November 30, 2016. Please write clearly and answer questions precisely. As a thumb rule, the answer should be limited to ≤ 2 written pages, with ample spacing between lines and in margins, per question. Always start a new question on a new page, starting with Question 1, followed by Question 2, ..., Question *n*. Please cite all the references (including web-sites, names of friends, etc.) which you have used/consulted as the source of information for each of the questions. BTW, when a question asks you to design an algorithm - it **requires** you to (1) Clearly spell out the **steps** of your algorithm in pseudocode (2) **Prove** that your algorithm is correct and (3) **Analyze** the running time. By default a graph G = (V, E) is simple, undirected and connected.

2 Problems

- 1. Consider the following version of the (weighted) planar-separator theorem. Let G = (V, E) be an embedded undirected triangulated planar graph, where n = |V|. Each vertex $v \in V$ has a positive weight $w(v) \ge 0$ and $\sum_{v \in V} w(v) = 1$. There exists a partition of V into disjoint sets A, B, and S, such that
 - (a) $w(A), w(B) \leq \frac{2}{3}$, where w(A) is the sum total of weights of all the vertices in set A
 - (b) $|S| \le 4\sqrt{n}$
 - (c) There is no edge in E that joins a vertex in A with a vertex in B.
 - (d) Such a set S can be found in linear time.

Show what changes you need to make in the proof of (unweighted) Planar Separator Theorem to prove the above theorem.

- 2. Prove the following: Let G = (V, E) be a connected planar graph and G^* be its dual. For any $E' \subseteq E$, the subgraph (V, E') has a cycle if and only if the subgraph $(V^*, E E')$ of G^* is disconnected.
- 3. Prove the following theorem on Geometric Separators. In 2-dimensions assume that you have n squares of arbitrary sizes. Squares are axis aligned. Moreover none of the points in the plane is inside more than k-squares. Prove that there exists either a vertical or a horizontal line which partitions the set of squares in such a way that at least $\lfloor \frac{n+1-k}{4} \rfloor$ of squares interiors

lie to each side of the line. How fast you can find such a line?

To start with first consider k = 1, and start with the problem in 1-dimension, each square is a segment, and the set of segments are disjoint. Then think about the case when k > 1 in 1-dimension ...

- 4. This problem is to show that an arbitrary range minima query (RMQ) problem can be solved within the same complexity as the one with the ± 1 RMQ problem. Recall that the ± 1 RMQ problem for an array of size n required O(n) time to preprocess and then the queries were answered in O(1) time. The idea is to reduce an arbitrary RMQ problem to the LCA problem. This reduction uses Cartesian Tree. Let A be an array consisting of n numbers (need not satisfy the ± 1 property). The Cartesian Tree C for A is defined as follows: The root of C is the minimum element of A, and it stores the position of this element in the array. Removing the root element splits the array into left and right subarrays. The left and right children of the root are recursively constructed Cartesian trees of the left and right subarrays, respectively. Prove the following:
 - (a) Cartesian tree C of an array A of size n can be computed in O(n) time (use incremental construction).
 - (b) Show that $RMQ_A(i,j) = LCA_C(i,j)$ (Recall that in C we store the indices i and j.)
- 5. Suppose we are given a flow network G, where edges have positive integer capacities, and $\mathcal{C} = \sum_{u,v \in V} c(u,v)$, where c(u,v) is the capacity of the edge $e = (u,v) \in E$. Show the following
 - (a) The value of the max flow is an integer.
 - (b) There is an assignment of non-negative integer flow values on each edge of G, satisfying all the flow conservation conditions, so that G achieves max flow.
 - (c) Show that the number of iterations required in the Ford-Fulkerson's algorithm (Residual network, find an augmenting path, augment the flow, repeat) is $O(\mathcal{C})$.
 - (d) Show that in the worst case, Ford-Fulkerson's algorithm, as stated in the Question 5c runs in exponential time.
 - (e) Construct an example, where one can realize the worst case as stated in the Question 5d.
- 6. Let G = (V, E) be the flow network. Let $C = \max_{(u,v) \in E} c(u, v)$ be the maximum capacity. Show the following:
 - (a) Minimum cut of G has a capacity of at most C|E|.
 - (b) For a given number K > 0, show how to find an augmenting path of capacity at least K in O(|E|) time, provided that such a path exists.
 - (c) Execute the following algorithm:
 - i. Initialize the flow f = 0;
 - ii. $K = 2^{\lfloor \log C \rfloor};$
 - iii. While $K \ge 1$ do

- A. While there exists an augmenting path p of capacity at least K then augment flow f along p.
- B. K := K/2;

iv. Return f.

Show that the above algorithm computes Max-Flow.

- (d) Show that the loop in Step iii. A is executed at most O(|E|) times for each value of K.
- (e) Show that the algorithm runs in $O(|E|^2 \log C)$ time.
- 7. A simple undirected graph G = (V, E) is called k-edge connected if removal of any set of k-edges keeps G still connected. (e.g. cycles are 1-edge connected.) Show how to compute edge connectivity of G by invoking at most |V| network flow computations.
- 8. Prove that the expected running time of finding the closest pair among a set of *n*-ponts in plane using the randomized incremental construction is O(n). (Refer to the talk on Closest Pairs.)
- 9. Given a well-separated pair decomposition for a set of *n*-points in plane,
 - (a) Show how you can determine the closest pair of points in O(n) time.
 - (b) Show how you can find an approximation to the diameter of the point set in O(n) time. What kind of approximation factor it will be in terms of the separation parameter s. (Diameter is the largest distance among the pairs.)
- 10. Let P be a set of n-points in plane. Define the complete graph G = (V, E) to be the graph where V = P, and there is an edge e = (uv) between each pairs of points $u, v \in P$. The weight of e is the Euclidean distance between u and v. Euclidean minimum spanning tree (EMST) of P is defined to be the minimum spanning tree of G. Show that using the well-separated pair decomposition, EMST can be approximated in $O(n \log n)$ time. Note that G has $\Omega(n^2)$ edges, and hence we cannot directly apply any of the minimum spanning tree algorithms.