Assignment 2

COMP 5703 - Fall 2016

November 8, 2016

1 Instructions

This is due at the start of the class on November 28, 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 an undirected graph $G = (V, E)$ on n vertices, where n is some power of 2. Let us label its vertices as $V = \{1, 2, 3, \ldots, n\}$. The edges of this graph are given by the following rules:

   (a) For each $1 \leq i \leq n-1$, $(i, i+1) \in E$.
   (b) For each $\sqrt{n}+1 \leq i \leq n$, $(i, i-\sqrt{n}) \in E$.
   (c) For each $1 \leq i \leq n-\sqrt{n}$, $(i, i+\sqrt{n}) \in E$.

   Does G contain a balanced $(\frac{1}{3} - \frac{2}{3})$ separator of size $O(\sqrt{n})$? Justify.

2. 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 $w(v) \geq 0$ 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) $|S| \leq 4\sqrt{n}$
   (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 the (unweighted) Planar Separator Theorem to prove the above theorem.

3. 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.

4. Prove the following theorem on Geometric Separators. In 2-dimensions assume that you have n squares of arbitrary sizes. Squares are axis aligned and they may overlap, but no point 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 ...

5. 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.)

6. Show that the Jaccard Distance which is defined as $1-$ {the Jaccard Similarity} between the two sets is a metric.

7. When applying amplification constructions to a locality-sensitive family of functions, we can apply an AND composition followed by ORs or vice-versa. Which order of composition is 'better', and why? Explain when you would apply AND followed by ORs, and when will you like to use ORs followed by ANDs.
   (Please have a look at the Chapter in my Notes on Locality Sensitive Hashing for this topic.)

8. Suppose we have a set P of n points in plane. We enclose these points in the smallest axis-aligned square box B. For the construction of quad-trees for a points in the plane, we can follow one of the following two recursive strategies:

   (a) Partition the box B recursive into 4 equal sub squares if it is non-empty.

   (b) Partition the box B by the following rule: Partition the non-empty box by a horizontal line or a vertical line (alternatively) passing through the median point. For example, in the first step, we partition B by the horizontal line passing through the point that has the median y-coordinate. This splits the box into two sub-boxes, say B_1 and B_2. In the next step each of the (non-empty) sub-boxes B_1 and B_2 are partitioned with respect to

the median vertical line among the points in the respective boxes. (For example, if B_1 is non-empty than first find the point $p \in P \cap B_1$ that has the median x-coordinate and then partition B_1 in two sub-boxes by a vertical line through p. Similarly, proceed with B_2 if $B_2 \cap P$ is non-empty. Next for each of the smaller non-empty boxes, we will first partition with respect to a horizontal line and so on.)

Suppose we want to use the quad-tree to report all the points in a query rectangle. Discuss the positive and negative aspects for each of the above two methods for answering such type of queries.

Next few problems relate to Fast Fourier Transforms.

9. Let $P(x)$ be a polynomial of degree n, i.e. $P(x) = a_0 + a_1x + a_2x^2 + \ldots + a_nx^n$. We will also assume that n is some power of 2 for simplicity. Let $x_0, x_1, \cdots x_n$ be $n+1$ distinct points (real or complex), and let $P(x_0), P(x_1), \ldots, P(x_n)$ be the value of the polynomial when evaluated at these points, respectively. That is, $P(x_j) = a_0 + a_1x_j + a_2x_j^2 + \ldots + a_nx_j^n$, for all $0 \leq j \leq n$.

   (a) Show that given the values $P(x_0), P(x_1), \ldots, P(x_n)$ corresponding to $n+1$ distinct points, the coefficients $a_0, a_1, \ldots, a_n$ can be determined. In other words, the polynomial $P(x)$ of degree n can be uniquely determined from its values at any $n+1$ distinct points.

   (b) Show that for any x_i, $P(x_i)$ can be evaluated in $O(n)$ time.

   (c) Let $E(x) = a_0 + a_2x + a_4x^2 + \ldots + a_nx^{\frac{n}{2}}$ and $F(x) = a_1 + a_3x + a_5x^2 + \ldots + a_{n-1}x^{\frac{n-2}{2}}$. Show that $P(x) = E(x^2) + xF(x^2)$.

   (d) Consider the evaluation of $P(x)$ at two different values z and $-z$. First, show that $P(z)$ can be computed in $O(n)$ time. Show that, given the computation of $P(z)$, $P(-z)$ can be computed in $O(1)$ time.

10. Let $\omega = e^{2\pi i/n}$, where $i = \sqrt{-1}$ is an imaginary number. Then $1, \omega, \omega^2, \ldots, \omega^{n-1}$ are the n-th complex roots of unity.

    (a) Show that $\omega^{n/2+j} = -\omega^j$, for $j = 0, 1, 2, \ldots, n/2 - 1$.

    (b) Show that the squares of each of $\{1, \omega, \omega^2, \ldots, \omega^{n-1}\}$ forms the $n/2$-th complex roots of unity.

    (c) Consider the evaluation of $P(x)$ for $x = 1, \omega, \omega^2, \ldots, \omega^{n-1}$, using the polynomials $E(x)$ and $F(x)$ as defined in the previous question. Can you design a recursive algorithm running in $O(n \log n)$ time to evaluate $P(x)$ for all these n-roots of unity. (It may be better to assume here that $P(x)$ is of degree $n-1$ instead of degree n, and $n-1$ is some power of 2, to keep the arguments simpler.)

    (d) Show that the above recursive method will not work if we choose $x = \pm 1, \pm 2, \ldots, \pm\frac{n}{2}$, instead of n roots of unity. What is the main reason it fails?