Final Assignment

COMP 5703 - Fall 2015

1 Instructions

This is due in SCS office (or my office) by 10 AM on Thursday December 17, 2015. 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. Show that the Jaccard Distance which is defined as $1-$ {the Jaccard Similarity} between the two sets is a metric.

2. (a) Prove that a matching is maximum if and only if there are no augmenting paths with respect to that matching.
   (b) Prove that a bipartite graph $G = (V = A \cup B, E)$ has a perfect matching if and only if for any subset $S \subseteq A$ the number of vertices adjacent to S in B (denote it by $N(S)$) must be as large as $|S|$ (i.e. $|N(S)| \geq |S|, \forall S \in A$).

3. Present a proof, in your own words, of the Isolating Lemma, (see the report on "Parallel Algorithm for Maximum Matching"). Where is it required in the parallel algorithm?

4. 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.

5. Let C be a circle, and let V be a set of n distinct vertices on its boundary. Form a maximal plane graph on V (i.e. we connect as many pairs of vertices as possible by straight line segments, so that no two edges cross each other in their interior). Notice that we obtain a plane triangulation of V. Call this triangulation X. Show that X has tree width of 2, and its tree decomposition can be computed in polynomial time. (Keywords: Triangulated simple polygon, dual graph, dual tree, tree-width)

6. Look at Procedure 1 in the report on Data Streams and Frequency Moment. Argue why 2^B is a good estimate of F_0.

7. Let $X = \{x_1, x_2, \cdots, x_n\}$ be a set of n-elements. Each element of x_i has a positive weight $w_i > 0$. Let $\mathcal{Y} = \{Y_1, Y_2, \cdots, Y_m\}$ be a set of subsets of X (i.e. each $Y_i \subset X$). A subset $H \subseteq X$ is called nice if $H \cap Y_i \neq \emptyset$, for $i = 1, \cdots, m$. The decision problem of finding a nice set of weight at most W is NP-Hard. Let W^* be the weight of the nice set with smallest possible weight. Let $\gamma = \max\{|Y_i|, i = 1, \cdots, m\}$. Provide an approximation algorithm, running in polynomial time, that computes a nice set whose weight is at most γW^*.

8. Given a simple graph $G = (V, E)$, we define a cut to be a partition of the vertex set V into two non-empty sets A and B, where $A \cup B = V$ and $A \cap B = \emptyset$. An edge $(a, b) \in E$ is said to cross the cut if $a \in A$ and $b \in B$. The size of the cut corresponding to the partition (A, B) is defined to be the number of edges crossing the cut. The maximum cut problem is to find a partition of V such that the size of the cut is maximized. Consider the following algorithm:
   Step 1: Find any partition of V.
   Step 2: For every vertex $v \in V$, if v would have more edges crossing the cut if placed in the opposite partition, then move v to the opposite partition.
   Prove the following

   (a) Prove that the above algorithm runs in polynomial time. What is the running time?

   (b) Prove that the size of the cut produced by the above algorithm is at least half of the size of the maximum cut. (In other words its an 1/2-approximation algorithm.)

9. Given a set P of points the plane. For each point $p \in P$ we denote by $b(p)$ the maximum Euclidean distance between p and any point in P, i.e., $b(p) = \max\{|pq| : q \in P\}$. We denote a point $p \in P$ with minimum $b(p)$ as the center of P. Let p be the center of P. Present a ~~polynomial~~ constant time algorithm that finds a point p' in P such that $b(p') \leq 2 \cdot b(p)$. Analyze the running time and prove the correctness of your algorithm.

10. Suppose you have a set S of n-points in the plane and you need to construct an approximate travelling salesperson tour $TSP(S)$ of S. All distances are measured with respect to Euclidean distance. You follow the following strategy. Choose any point $s \in S$, and initialize a trivial tour $T = \langle ss \rangle$. Now we will grow this tour. Find a point $v \in S \setminus \{s\}$ that is closest to s, and update the tour to include v and the current tour becomes $T = \langle svs \rangle$. In general, suppose currently our tour consists of $k + 1$ vertices $T = \langle su_1u_2...u_ks \rangle$. Now find a vertex in $v \in S$ that is closest to (but distinct from) $s, u_1, u_2, ..., u_k$. Let v be closest to $u \in \{s, u_1, u_2, ..., u_k\}$. Then the new tour is obtained by inserting v just after u in T. (For example, if v was closest to u_3, than the new tour will be $T = \langle su_1u_2u_3vu_4...u_ks \rangle$.) We repeat this process till all the points in S are added to T. Show that the cost of T is at most twice the cost of an optimal tour. (Note that the cost of a tour is the sum total of the costs of all the edges in the tour.)