15-451 Algorithms, Spring 2012

Homework 5 (100 pts) Due: Mar. 23-25

oral presentation

Ground rules:

- This is an oral presentation assignment. You should work in groups of two or three. At some point your group should sign up for a 1.5-hour time slot on the signup sheet on the course web page.
- Each person in the group must be able to present every problem. The TA/Professor will select who presents which problem. The other group members may assist the presenter.
- You are not required to hand anything in at your presentation, but you may if you choose. If you do hand something in, it will be taken into consideration (in a non-negative way) in the grading.

Question	Points	Score
1	50	
2	20	
3	30	
Total:	100	

1. **Low Stretch Trees**

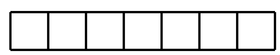
One of the most fundamental types of graphs is the spanning tree. In class we considered the maximum/minimum weight spanning tree. In the case when all edges of the graph have the same weight then all spanning trees have the same weight. The goal of this problem is to define another class of spanning trees, namely, ones called **low stretch spanning trees**. Recall that in a tree there is a unique path connecting any two nodes and we define the length of this path to be the number of edges on this path.

Let $G = (V, E)$ be a connected, unweighted and undirected graph with n vertices and m edges and $T = (V, E')$ a spanning tree of G. If $e = (v, w)$ is an edge of G we define the **stretch** of e in T to be the length of the path in T from v to w, denoted by $Str_T(e)$. The **stretch of T in G** is:

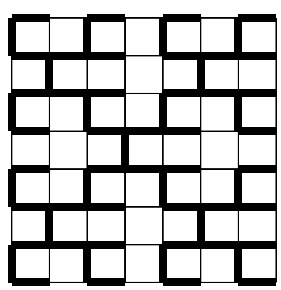
$$Str(T, G) = \sum_{e \in E} Str_T(e)$$

The average stretch is $Str(T, G)/m$

(5) (a) Construct two spanning trees T_1 and T_2 for the 2 by n mesh graph with average stretch $\Theta(1)$ and $\Theta(n)$, respectively. The 2 by 8 mesh graph is shown below.



(10) (b) Show that the recursive "C" construction for the $\sqrt{n}$ by $\sqrt{n}$ mesh has average stretch $O(\log n)$. Here you may assume that $\sqrt{n}$ is a power of 2. Below is an example of the tree for the 8 by 8 square mesh.



The goal of the next few parts of this problem is to find a linear time algorithm that finds a spanning tree with average stretch $O(\sqrt{n}\log n)$. We make a few definitions. Let B be subset of vertices define $E(B)$ to be all the edges with both endpoints in B and ∂B to be all the edges with exactly one endpoint in B. Let $B_i(s)$ be all the vertices of V at a distance at most i in G from vertex s.

(10) i. Show that for some $1 \leq i \leq \sqrt{n}\log m$ it will be the case that

$$|\partial B_i| \leq |E(B_i)|/\sqrt{n} \tag{1}$$

Hint: Proof by contradiction. Also recall the important fact from your calculus class

$$2 < (1+1/x)^x < e \quad \text{for} \quad 2 < x$$

(15) ii. Give an algorithm to partition the vertices of G into vertex disjoint sets $S_1, \ldots, S_k$ such that

1. The induced graph on each S_i is connected, i.e., $H_i = (S_i, E(S_i))$ is a connected graph.
2. The diameter of each H_i is at most $\alpha = 2\sqrt{n}\log m$, i.e., the maximum distance between any two nodes in H_i is at most α.
3. $|\partial H_i| \leq |E(H_i)|/\sqrt{n}$

Your algorithm should run in $O(n+m)$ time. Hint: Use the result from part i.

(10) iii. Use the decomposition from part ii to construct your tree with average stretch $O(\sqrt{n}\log n)$.

2. **Common Intersection**

Suppose we have n triangles $T_1, \ldots, T_n$ in the plane. The goal is determine if the n triangles have a common intersection, that is, does there exist a point $p \in \cap_{i=1}^{n} T_i$.

(10) (a) Give a randomized algorithm the determines if the triangles $T_1, \ldots, T_n$ have a common intersection.

(10) (b) Give an $O(n \log n)$ algorithm that computes the area of the intersection.

3. **Tsunami High-ground Problem.**

   In this problem we are given a convex polygon where the points/corners are $P = (p_1, \ldots, p_n)$ in counterclockwise order.

(10) (a) Give a expected linear time algorithm to find a point p such that if we remove any k of the points from P the point p will still be in convex closure of the remaining points.

(10) (b) Give a deterministic linear time algorithm for the problem in part (a) Note: To get full credit for this part and the last part you should give two quite different algorithms for each part.

(10) (c) Give an $O(n \log n)$ expected time algorithm that finds a point p corresponding to the largest possible value for k.