6.856 — Randomized Algorithms

David Karger

Handout #19, November 8, 2002 — Homework 10, Due 11/13

1. A *flow* in an undirected graph is a set of edge-disjoint paths from a *source* vertex s to a *sink* vertex t. The *value* of the flow is the number of edge disjoint paths. The *s-t maximum flow problem* aims to a flow of maximum value. This quantity turns out to be equal to the *s-t minimum cut value:* the minimum number of edges that must be removed from the graph in order to disconnect vertex s from vertex t. There is an *augmenting path algorithm* that, given an s-t flow of value v, finds an s-t flow of value $v+1$ in $O(m)$ time on an m-edge graph, or else reports that v is the maximum flow.

   Consider any undirected graph with m edges, s-t maximum flow v, and minimum cut c:

   (a) Prove for any constant ϵ, an s-t cut of value at most $(1+\epsilon)v$ can be found in $\tilde{O}(mv/c^2)$ time.

   (b) Prove that for any constant ϵ, a flow of value $(1-\epsilon)$ can be found in $\tilde{O}(mv/c^2)$ time.

   (c) Sketch an algorithm that finds the maximum flow in $\tilde{O}(mv/\sqrt{c})$ time, and give an *informal argument* as to its correctness.

   (d) **Optional.** Use the algorithm of part (c) to improve the running times of the algorithms in parts (a) and (b)

2. Consider the problem of finding the smallest (minimum diameter) circle containing some set H of n points in the plane. We will assume that the points are in "general position"—no 3 points are colinear, and no 4 points are on the boundary of a common circle. This assumption can be achieved by small perturbations in the input. For any set of points S in the plane, let $O(S)$ denote the smallest circle containing S.

   (a) Show that for any 3 non-colinear points, there is a unique circle having all 3 of those points on the circle boundary. This circle (center and radius) can be computed in constant time from the points.

   (b) Show that $O(H)$ contains either 2 or 3 of the input points on its boundary. We will call these points the "basis" of the circle (hint, hint) and refer to them as $B(H)$. Deduce a simple $O(n^4)$-time algorithm for solving the problem.

   (c) Show that if a circle C excludes a point of H, then C cannot be the smallest circle containing B(H).

(d) Show that if p is *not* contained in $O(S)$ for some S then p is on the boundary of $O(S \cup \{p\})$.

(e) Consider a set R of r points chosen at random from H. Bound the expected number of points of H outside $O(R)$.

(f) Generalize the previous part to where you have an "active" subset $S \subseteq H$ and compute $O(R \cup S)$.

(g) Give an $\tilde{O}(n)$ time algorithm for finding $O(H)$.