

## 6.856 — Randomized Algorithms

David Karger

Handout #18, November 2, 2002 — Homework 8 Solutions

**Problem 1** First let's eliminate edges of length 0. Two vertices connected by an edge of length 0 have the same distance to everything, so we can contract all such edges and work in the simplified graph without 0-edges. Let  $D(i, j)$  be the distance from  $i$  to  $j$ . We know that

1. every neighbor  $k$  of  $i$  has  $D(i, j) - B \leq D(k, i) \leq D(i, j) + B$
2. some neighbor has  $D(i, k) + D(k, j) = D(i, j)$ .

So suppose we consider taking distances modulo  $2B+1$ . If  $D(i, j) = d$  and  $D(i, k) + D(k, j) \equiv d \pmod{2B+1}$ , then  $D(i, k) + D(k, j) = d + m(2B+1)$ . But the first item above shows that  $m$  must be zero, meaning any such  $k$  is on the shortest path from  $i$  to  $j$ . Such a neighbor must exist by item 2 above.

So consider the matrix  $A^{(d)}$  where  $A^{(d)}(i, k)$  is 1 if  $D(i, j) \equiv d \pmod{2B+1}$  and 0 otherwise. If  $D(i, j) \equiv d \pmod{2B+1}$  then we could find a successor for  $i$  on the shortest  $(i, k)$  by finding a (known to exist) vertex  $k$  such that for some  $r$ ,  $D(i, k) = r$  and  $D(k, j) = d - r$ . Finding such a vertex requires checking the  $(i, j)$  entry in a WBPM of  $A^{(r)}$  and  $A^{(d-r)}$ . We need to do this for  $B$  possible values of  $r$  and  $2B+1$  possible values of  $d-r$ , so the total work is  $B^2$  WBPM computations.

**Problem 2** The key step is to modify the weights so that the minimum weight perfect matching is unique; at this point the in-class construction can be applied, so long as all weights are polynomial.

Consider the initial graph, and let  $W$  be the minimum weight of a perfect matching. As a first step, we multiply all edge weights (and thus all perfect-matching weights) by  $3nm$ . In the new graph the minimum weight matching has value  $3nmW$ , while the next-smallest weight of a matching (originally at least  $W+1$ ) is at least  $3nm(W+1) = 3nmW + 3nm$ . Suppose we now add an independent random integer perturbation, between 1 and  $2m$ , to each edge. Note that this changes the value of each perfect matching by at most  $n(2m) = 2nm$ . It follows that any originally minimum-weight matching ends up with weight at most  $n^3W + 2nm$ . Since any originally non-minimum weight matching has weight at least  $n^3W + 3nm$ , it follows that any minimum weight matching in the perturbed graph corresponds to a minimum weight matching in the original graph.

It remains to show that the minimum weight matching in the perturbed graph is unique. To do so, consider all edge sets which correspond to originally-min-weight perfect matchings.

Each such set receives a “total perturbation” equal to the sum of perturbations on its edges. By the isolating lemma, since there are only  $m$  distinct edges to be perturbed, the probability that the minimum total perturbation is unique is at least  $1/2$ . Since all min-weight matchings start with the same original weight, if the minimum total perturbation is unique, it follows that we will end up with a unique min-weight perfect matching.

**Problem 3** As for the in-class algorithm, we focus on *good* vertices, those vertices  $v$  with at least  $d(v)/3$  neighbors of degree no more than  $d(v)$ .

Note first that the probability that two vertices get the same weight is equal to the probability of a collision when  $n$  vertices are mapped to  $n^4$  weights, which is less than  $1/n^3$ . So we can assume that no collision happens, and perform the rest of our analysis conditioned on the event of no collisions. Conditioned on this event, all that is important in the analysis is the relative (ranking) order of weights on the nodes. It is easy to see by symmetry that this random ranking is a uniform permutation of the vertices.

The easiest way to analyze this problem is to map it back to the in-class analysis. Using the specified weight assignment, declare a vertex  $w$  to be *marked* if its assigned weight is at most  $n^4/2d(w)$ . Note that this happens with probability  $1/2d(w)$ . Note also that the events of different vertices being marked are *independent* since each vertex’s marking depends only on the weight of that vertex.

Now let us consider a particular *good* vertex  $v$ , i.e. one with at least  $d(v)/3$  neighbors of degree at most  $d(v)$ . Such a neighbor  $w$  is marked with probability  $1/2d(w) \geq 1/2d(v)$ . Since there are at least  $d(v)/3$  of these low-degree neighbors, and since the marking events are independent, the probability that *no* such neighbor of  $v$  is marked is at most  $(1 - 1/2d(v))^{d(v)/3} \leq e^{-1/6}$ .

Conditioned on a vertex being marked, let us determine the probability it is incorporated in  $S$ . This happens if no neighbor of  $w$  gets a smaller weight. Since the weight on  $w$  is at most  $n^4/2d(w)$ , the probability that any particular neighbor gets a smaller weight is at most  $1/2d(w)$ . Since there are only  $d(w)$  neighbors, it follows from the union bound that, conditioned on  $w$  being marked, the probability it has a smaller-weight neighbor is at most  $1/2$ . This implies that any marked vertex is included in  $S$  with probability at least  $1/2$ .

Combining these two arguments, exactly as in the text, we deduce that some neighbor of a good vertex is incorporated into  $S$  with probability at least  $e^{-1/6}/2$ . The remainder of the analysis (which is based on bounding the expected number of good edges removed in an iteration) proceeds exactly as in the text.

**Problem 4** Most correct solutions got  $W = O(n^6)$  using the Isolating Lemma. Here we present a neat solution achieving  $O(n^5)$  due to Harold Fox. The solution actually “sacrifices” some randomness in order to make the analysis easier and get the better bound.

Find a MST of the graph. Fix every edge not in the MST at a length  $\Theta(n^5)$ . Randomly assign the lengths to the edges of the MST from  $1 \dots n^4$ . Note that this assignment of lengths guarantees that the shortest paths will only use edges of the MST. Furthermore, since shortest paths in a MST (or any tree for that matter) are unique, our first condition is always satisfied. Now we consider the second condition that all pairwise distances are distinct. Consider any pair of pairs  $(i,j), (k,l)$ . By our construction, they have different distances if the random

assignment of lengths to the edges of the 2 paths  $i$  to  $j$  and  $k$  to  $l$  make them different. Since there must be one edge in one and not the other, the probability of equal distances is  $< \frac{1}{n^4}$ . Union bounding over  $\binom{n}{2}$  pairs of pairs gives a probability still less than 1, and so the Probabilistic Method guarantees the existence of a good assignment. Thus,  $W$  need only be  $O(n^5)$ .