Matthew Feng December 5, 2018

6.046 Problem Set 10

Collaborators: *None*

Problem 1

(a)

We note that the maximum of routers that may remain on in any 4×4 grid, no matter how many routers are available, is 9.

Thus, we only need to check all subsets of the n routers that are size 9 or less. There are $O(\binom{n}{9})$ possible subsets, and for each subset we need to do maximum of $O(n^2)$ work to check if any pairs of points intersect, which yields in total a polynomial time algorithm in the number of routers.

For the later parts, we will adjust this thinking to considering routers that are completely contained within the 4×4 area; in that case, a maximum of 4 routers be on at once in a 4×4 area. However, the algorithm is still polynomial in n because the size of the subsets we need to enumerate is bounded.

(b)

The divide and conquer algorithm doesn't work because the optimal set of routers may overlap blocks of 4×4.

In the first algorithm method, since we may select routers that overlap into other grid cells, we may force out the selection of some optimal routers based on the order we iterate over the grid cells (or if we don't bother checking router coverage, the router coverage will overlap).

In the second algorithm method, if we only consider routers completely contained within a grid cell, then potentially optimal routers that cross grid cells will not be selected.

(c)

We can take the approach of the second algorithm, that is only selecting routers that are completely contained within a grid cell, and combine it with a randomly offset grid of width 4 along both x and y axes, such that the expected number of routers that are selected to remain on is $\frac{|C^*|}{4}$, where C^* is the optimal set of routers.

To see this, consider the optimal set C^* of routers. With a randomly offset grid, the probability that a router $c \in C^*$ has its unit disk of coverage completely contained within a grid cell of width 4 is $\frac{1}{4}$, since both the x and y axes must not intersect the disk. In particular, suppose the disk is centered at $(1, 1)$. For a grid of width 4, the only offsets that don't result in a grid intersecting with the disk are within the box bounded by $(2, 2)$ and $(4, 4)$ (since the grid repeats, we don't need to consider any other offsets). Thus, a randomly offset the expected number of disks in the optimal set lie entirely within a grid cell is $\frac{|C^*|}{4}$ by linearity of expectation (and indicator variables), as desired.

(d)

We can generalize this algorithm into a PTAS by increasing the size of our grid cells. Suppose we have a grid where the width of each cell is k. Then the probability that a unit disk is not intersected by a randomly offset grid is

$$\frac{(\frac{k}{2} - 1)^2}{(\frac{k}{2})^2} = (1 - \frac{2}{k})^2.$$

Thus, the expected fraction of the optimal solution we will achieve is $(1 - \frac{2}{k})^2$. Setting $1 - \epsilon = (1 - \frac{2}{k})^2$. Solving for k yields

$$k = \frac{2}{1 - \sqrt{1 - \epsilon}}.$$

With at most $(\frac{k}{2})^2$ routers per cell, the running time of any algorithm in the PTAS is $O(n^k)$ (since there are constant number of grid cells, and the maximum size of a subset for a cell is $(\frac{k}{2})^2$)).

Problem 2

(a)

With a single query, the case where i and j have not spoken with each other may be attributed to one of two reasons:

1. i and j are in the same agency.
2. i and j are in different agencies, but by chance are not in the $\frac{3n}{4}$ spies that spoke to each other.

Since we cannot distinguish between to the two reasons with a single query, and the two reasons yield opposite answers, we cannot be correct on all inputs.

(b)

We can conclude that spies i and j are from the same agency, since they cannot be in the agencies of m and n whom are part of different agencies, and there are only three agencies in total.

(c)

The probability that m and n, sampled uniformly at random, are in different agencies, are not in the same agency as i and j, have talked to each other, and have each spoken with i and j, given that i and j are in the same agency, is at least

$$(\frac{2}{3})^3 \times (\frac{3}{4})^5 = \frac{9}{128} > \frac{1}{36},$$

because there is

1. 2/3 probability m and n are in different agencies,
2. 2/3 probability m and i are in different agencies (which implies m and j as well),
3. 2/3 probability n and i are in different agencies,
4. at least 3/4 probability for each pair in $\{(m,n),(m,i),(m,j),(n,i),(n,j)\}$ to have spoken with each other.

(d)

Given two spies i and j, there are two cases which we need to consider:

Case 1. i **and** j **have talked to each other.** In this case, i and j must be in different agencies, so we output `DIFFERENT`.

Case 2. i **and** j **have** *not* **talked to each other.** In this case, we use the construction described in part (b) which samples two other spies m and n uniformly at random to see if i and j are part of the same agency. With probability of at least $\frac{1}{36}$, we will find the construction, letting us conclude that i and j are in the same agency, and we output `SAME`. Otherwise, we output `DIFFERENT`.

Our algorithm has the desirable property that if i and j are in different agencies, then we will always output the correct answer `DIFFERENT`, since the construction leading the output of `SAME` is impossible to sample. However, if i and j are in the same agency, there is at least a $\frac{1}{36}$ probability that we will output the correct answer.

With this property (that the output for one possible outcome is always correct, and the other is probabilistically correct), we can simply run this algorithm many times until we get the desired accuracy.

In particular, instead of sampling (m, n) only once, we sample k times. Each time, there is at most $\frac{35}{36}$ probability that the construction does not exist, but i and j are in the same agency. There is only a probability of at most $(\frac{35}{36})^k$ for this to occur all k times (we know that if the construction exists for any of the k samples, i and j are in the same agency). Thus, the probability of outputing the incorrect answer is $(\frac{35}{36})^k$. For $k \geq 25$, the probability of incorrectness is below 1/2 (i.e. probability of correct output is greater than 1/2, as desired). k is a constant, and checking to see if the construction in (b) exists takes a constant number of queries, which means we can determine whether or not i and j belong to the same agency with a constant number of queries.