COMS 4995 (Randomized Algorithms): Exercise Set #7

For the week of October 14–18, 2019

Instructions:

(1) *Do not turn anything in.*

(2) The course staff is happy to discuss the solutions of these exercises with you in office hours or in the course discussion forum.

(3) While these exercises are certainly not trivial, you should be able to complete them on your own (perhaps after consulting with the course staff or a friend for hints).

Exercise 31

Recall the disjoint paths problem from Lecture #12. The input is a graph $G = (V, E)$ (directed, let's say) and k source-sink pairs $(s_1, t_1), \ldots, (s_k, t_k)$. The goal is to choose one s_i-t_i path P_i for each source-sink pair such that no edge is used in more than one of the paths (or to determine that this cannot be done). Show that when $s_1 = s_2 = \cdots = s_k$ and $t_1 = t_2 = \cdots = t_k$, the problem can be solved in polynomial time.

[Hint: Reduce to the maximum flow problem.]

Exercise 32

Suppose we modify the disjoint paths problem in the preceding exercise by allowing up to $u := c\frac{\ln m}{\epsilon^2}$ paths to use each edge, where m is the number of edges in the graph, $\epsilon \in (0, 1)$ is a parameter, and c is a sufficiently large constant (independent of m and ϵ). Modify the analysis of the randomized rounding algorithm from Lecture #12 to show that the algorithm either correctly deduces that the problem is infeasible, or else (with high probability) it outputs one path per s_i-t_i pair such that no edge is used by more than $(1 + \epsilon)u$ of the paths.

Exercise 33

Consider a random graph G drawn from the Erdös-Renyi distribution $\mathcal{G}(n, \frac{1}{2})$ described in Lecture #12, with n even. (I.e., the original version of the model, not the version with the planted bisection.) Prove that, for every $\epsilon \in (0, 1)$, the number of edges crossing every bisection of G is between $(1 - \epsilon)\frac{n^2}{8}$ and $(1 + \epsilon)\frac{n^2}{8}$ with probability $1 - o(1)$, where the $o(1)$ term tends to 0 as $n \to \infty$.[1]

Thus, in uniformly random graphs, even an algorithm that computes a *maximum* bisection (with the largest number of crossing edges) is a good approximation algorithm for the problem of computing a minimum bisection!

[Hint: this follows from the classic one-two punch of the Chernoff bound and the Union bound.]

Exercise 34

Recall the one-way communication complexity model from Lecture #13. Recall EQUALITY, which is the function $f : \{0, 1\}^n \times \{0, 1\}^n \to \{0, 1\}$ given by $f(x, y) = 1$ if $x = y$ and $f(x, y) = 0$ if $x \neq y$. Prove that every deterministic one-way protocol that correctly decides EQUALITY uses n bits of communication (from Alice to Bob) in the worst case.

[1] A *bisection* is a cut (A, B) of the graph in which A and B have equal sizes.

[Hint: If Alice always sends at most $n-1$ bits to Bob, she must send the same message z for two different inputs x, x' (why?). What does Bob do if he receives this message z from Alice and his input y is the same as x or x'?]

Exercise 35

Prove that if x, y are distinct n-bit strings and r is a uniformly random n-bit string, then there is a 50% chance that the inner products $\langle x, r\rangle$ and $\langle y, r\rangle$ differ modulo 2.

[Hint: Focus on a coordinate where x and y differ, and use the Principle of Deferred Decisions.]