Assignment 2: Randomized Algorithm Design

Palash Dey
Indian Institute of Technology, Kharagpur

February 12, 2020

1. Suppose we have a Monte Carlo randomized algorithm $\mathcal{A}$ for some decision problem Π which, on every input x, outputs correct answer with probability at least $3/4$. Suppose $\mathcal{A}$ uses $\mathcal{O}(\log n)$ random bits where n is the size of input. Prove that there exists a deterministic polynomial time algorithm for the problem Π.

2. In the Exact 3SAT problem, the input is m clauses where every clause is OR of 3 different literals over n variables and the goal is to find an assignment of the variables which makes maximum number of clauses TRUE. Consider the following simple randomized algorithm. Set every variable TRUE with probability $1/2$ and FALSE with probability $1/2$. Prove that the expected number of clauses satisfied by the algorithm above is $\frac{7m}{8}$. Derive concentration bound for this algorithm using Chebyshev inequality.

3. **[Permutation Routing on Hypercube]** Let n be a positive integer and $\mathcal{N} = 2^n$. An hypercube is a graph where we have a vertex v_x for every $x \in \{0, 1, \ldots, \mathcal{N} - 1\}$. Hence the total number of vertices is $\mathcal{N}$. For any two vertices v_x and v_y, we have an (undirected) edge between v_x and v_y if and only if the Boolean representation of x and y differs in exactly one bit. In the permutation routing problem, every vertex in an hypercube has exactly one packet to send to some other vertex and receives exactly one packet. Hence, the problem corresponds to a permutation $\pi : \{0, 1, \ldots, \mathcal{N}-1\} \longrightarrow \{0, 1, \ldots, \mathcal{N}-1\}$ where vertex x sends a packet to $\pi(x)$ and receives a packet from $\pi^{-1}(x)$.

   Every node can send and receive simultaneously. However, at each time step (time is discrete for this problem), at most one packet can be sent along one edge. Hence some vertex wants to send two packets along the same edge, one packet has to wait till next time step. The objective of the permutation routing problem is to route all the packets in minimum number of time steps.

   A natural algorithm for this problem is *bit fixing*. Consider a packet that needs to be sent from v_x (its current location) to v_y. Let the Boolean representation of x and y be $x_{n-1}x_{n-2}\ldots x_1x_0$ and $y_{n-1}y_{n-2}\ldots y_1y_0$. We find the minimum i such that $x_i \neq y_i$ (such an i is guaranteed to exist if $x \neq y$). Let x' be $(x_{n-1}x_{n-2}\ldots x_{i+1}y_ix_{i-1}\ldots x_1x_0)_2$. Then the packet is sent along the edge $\{v_x, v_{x'}\}$ (of course it may have to wait if there more packets wanting to use this edge at the same time). Observe that the routing algorithm does not consider other packets for deciding its route. Such routing algorithms are called *oblivious* routing algorithm. Then prove the following.

   (a) **[Bit fixing is not good in worst case]** Give an instance of the problem where bit fixing algorithm takes $\Omega(\sqrt{\mathcal{N}}/n)$ time steps[1].

   (b) **[Random intermediate destination]** The idea to bypass worst case instances is to first route packets to random intermediate instances and then route them to destinations (you have already seen this idea in the first assignment isn't it?). Formally, let $\sigma : \{0, 1, \ldots, \mathcal{N}-1\} \longrightarrow \{0, 1, \ldots, \mathcal{N}-1\}$ be a random permutation. For every $x \in \{0, 1, \ldots, \mathcal{N} - 1\}$, we first route the packet for x form v_x to $v_{\sigma(x)}$ and then from $v_{\sigma(x)}$ to $v_{\pi(x)}$. Prove that, on every instance, the expected number of time steps that the modified algorithm takes is $\mathcal{O}(n)$.

[1] Actually, it is known that any deterministic oblivious routing protocol takes $\Omega(\sqrt{\mathcal{N}}/n)$ time steps in the worst case.

4. Let $\mathcal{P}$ and $\mathcal{Q}$ be two distribution over the σ-algebra $([n], 2^{[n]})$. Then prove the following.

$$2d_{TV}(\mathcal{P}, \mathcal{Q}) = ||\mathcal{P} - \mathcal{Q}||_1$$

5. Prove that the randomized algorithm for 2SAT when applied to 3SAT (pick an unsatisfied clause and complement a variable appearing in it) takes $\Omega(2^n)$ steps to discover a satisfying assignment if there is any.

6. A random walk on a connected undirected graph is aperiodic if and only if the graph is not bipartite.

7. Show that the mixing time of a random walk on an n dimensional hypercube is at most $n \ln n + n \ln(1/\varepsilon)$.

8. Give an example of a DNF formula with m clauses and n variables with constantly many (at least 1) satisfying assignments. Prove that the probability that at least one of the $2^{n/2}$ uniformly random assignments satisfies the formula is $\Omega\left(\frac{1}{2^{n/2}}\right)$.