CS612 Assignment 9

Institute of Computing Technology,
Chinese Academy of Sciences, Beijing, China

December 22, 2009

Notice:

1. Due Jan. 7, 2010.
2. Please send your answer to wangchao1987@ict.ac.cn, shaomingfu@gmail.com, yuanxiongying@ict.ac.cn
3. You can arbitrarily choose two problems from Problems 1-7.

1   Randomized Algorithm(5 marks)

$3 - Coloring$ is a yes/no question, but we can phrase it as an optimization problem as follows.

Suppose we are given a graph $G = (V, E)$, and we want to color each node with one of three colors, even if we aren't necessarily able to give different colors to every pair of adjacent nodes. Rather, we say that an edge (u, v) is $satisfied$ if the colors assigned to u and v are different.

Consider a 3-coloring that maximizes the number of satisfied edges, and let c^* denote this number. Give a polynomial-time algorithm that produces a 3-coloring that satisfies at least $\frac{2}{3}c^*$ edges. If you want, your algorithm can be randomized; in this case, the $expected$ number of edges it satisfies should be at least $\frac{2}{3}c^*$.

2   Randomized Algorithm(5 marks)

Consider a county in which 100000 people vote in an election. There are only two candidates on the ballot: a Democratic candidate(denoted D) and a Republican candidate(denoted R). As it happens, this county is heavily Democratic, so 80000 people go to the polls with the intention of voting for D, and 20000 go to the polls with the intention of voting for R.

However, the layout of the ballot is a little confusing, so each voter, independently and with probability $\frac{1}{100}$, votes for the wrong candidate, that is, the one that he or she didn't intend to vote for.

Let X denote the random variable equal to the number of votes received by the Democratic candidate D, when the voting is conducted with this process of error. Determine the expected value of X, and give an explanation of your derivation of this value.

3 Randomized Algorithm(5 marks)

Consider the following simple model of gambing in the presence of bad odds. At the beginning, your net profit is 0. You play for a sequence of n rounds; and in each round, your net profit increases by 1 with probability $\frac{1}{3}$, and decreases by 1 with probability $\frac{1}{3}$.

Show that the expected number of steps in which your net profit is positive can be upper-bounded by an absolute constant, independent of the value of n.

4 Randomized Algorithm(10 marks)

Consider the following analogue of Karger's algorithm for finding minimum $s-t$ cuts. We will contract edges iteratively using the following randomized procedure. In a given iteration, let s and t denote the possibly contracted nodes that contain the original nodes s and t, respectively,. To make sure that s and t do not get contracted, at each iteration we delete any edges connecting s and t and select a random edge to contract among the remaining edges. Give an example to show that the probability that this method finds a minimum $s-t$ cut can be exponentially small.

5 Randomized Algorithm(10 marks)

Consider a balls-and-bins experiment with $2n$ balls but only two bins. As usual, each ball independently selects one of the two bins, both bins equally likely. The expected number of balls in each bin is n. In this problem, we explore the question of how big their difference is likely to be. Let X_1 and X_2 denote the number of balls in the two bins, respectively. Prove that for any $\epsilon > 0$ there is a constant $c > 0$ such that the probability $\Pr(X_1 - X_2 > c\sqrt{n}) \leq \epsilon$.

6 Randomized Algorithm(10 marks)

Suppose you are presented with a very large set S of real numbers, and you'd like to approximate the median of these numbers by sampling. You may assume all the numbers in S are distinct. Let $n = |S|$; we will say that a number x is an $\epsilon - approximate\ median$ of S if at least $(\frac{1}{2} - \epsilon)n$ numbers in S are less than x, and at least $(\frac{1}{2} - \epsilon)n$ numbers in S are greater than x.

Consider an algorithm that works as follows. You select a subset $S' \subset S$ uniformly at random, compute the median of S', and return this as an approximate median of S. Show that there is an absolute constant c, independent of n, so that if you apply this algorithm with a sample S' of size c, then with probability at least 0.99, the number returned will be a 0.05-approximate median of S.

7 Randomized Algorithm(10 marks)

Some people designing parallel physical simulations come to you with the following problem. They have a set P of k *basic processes* and want to assign each process to run on one of two machines, M_1 and M_2. They are then going to run a sequence of n *jobs*, $J_1, ..., J_n$. Each job J_i is represented by a set $P_i \subset P$ of exactly $2n$ basic processes which must be running while the job is processed. An assignment of basic processes to machines will be called *prefectly balanced* if, for each job J_i, exactly n of the basic processes associated with J_i have been assigned to each of the two machines. An assignment of basic processes to machines will be called *nearly balanced* if, for each job J_i, no more than $\frac{4}{3}n$ of the basic processes associated with J_i have ben assigned to the same machine.

(a)Show that for arbitrarily large values of n_i there exist sequences of jobs $J_1, ..., J_n$ for which no perfectly balanced assignment exists.

(b)Suppose that $n \geq 200$. Give an algorithm that takes an arbitrary sequence of jobs $J_1, ...J_n$ and produces a nearly balanced assignment of basic processes to machines. Your algorithm may be randomized, in which case its expected running time should be polynomial, and it should always produce the correct answer.