CS 577: Introduction to Algorithms 11/13/2012

Homework 5

Instructors: Shuchi Chawla and Dieter van Melkebeek Due: 11/29/2012

This assignment covers randomized algorithms (chapter 13 in the textbook). See the first assignment for guidelines.

Ungraded problems

1. The bias of a coin is the probability of obtaining heads in a single toss of the coin. In algorithm design, we are often interested in coins of bias 1/2, but natural sources of randomness (such as radioactive decay) can be quite skewed.

   (a) Given a coin C of bias p, give an algorithm for generating a coin toss with bias 1/2, using as few tosses of C in expectation as possible. Can you design an algorithm that **does not** require knowledge of p?

   (b) Sometimes we have available a coin of bias 1/2, but require a coin of bias p for some value $p < 1/2$. Give an algorithm for generating a coin toss of bias p using a coin C with bias 1/2 using as few tosses of C in expectation as possible.

2. Problem 13.12 in the textbook (Pg. 790).

3. Problem 13.9 in the textbook (Pg. 788-789). Note that it is possible to get a success probability better than 1/4 but you only need to prove a 1/4 for this homework.

Graded problems

4. (5 points) You are given n nuts and n bolts of different sizes. Each nut fits exactly one bolt. Your job is to pair up the nuts with their matching bolts. You are allowed to compare nuts to bolts by trying to fit one into the other. The result of such a comparison is that the nut fits exactly with the bolt, or the nut is bigger, or the bolt is bigger. You cannot compare nuts to nuts or bolts to bolts. Give a randomized algorithm for solving this problem that runs in time $O(n \log n)$ in expectation.

5. (5 points) Problem 13.8 in the textbook (Pg. 788).

6. (5 points) You are given n points each picked independently uniformly at random from the two-dimensional unit square $[0, 1]^2$. Each point can be described by its x-coordinate and its y-coordinate. We say that a point (x_1, y_1) dominates another one (x_2, y_2), if $x_1 > x_2$ and $y_1 > y_2$. A point is undominated if no other point dominates it. Compute the expected number of undominated points.

Extra-credit problems

7. Give a polynomial time algorithm for the attached ICPC problem "The Lost House".

3141 - The Lost House

Asia - Beijing - 2004/2005

One day a snail climbed up to a big tree and finally came to the end of a branch. What a different feeling to look down from such a high place he had never been to before! However, he was very tired due to the long time of climbing, and fell asleep. An unbelievable thing happened when he woke up he found himself lying in a meadow and his house originally on his back disappeared! Immediately he realized that he fell off the branch when he was sleeping! He was sure that his house must still be on the branch he had been sleeping on. The snail began to climb the tree again, since he could not live without his house.

When reaching the first fork of the tree, he sadly found that he could not remember the route that he climbed before. In order to find his lovely house, the snail decided to go to the end of every branch. It was dangerous to walk without the protection of the house, so he wished to search the tree in the best way.

Fortunately, there lived many warm-hearted worms in the tree that could accurately tell the snail whether he had ever passed their places or not before he fell off.

Now our job is to help the snail. We pay most of our attention to two parts of the tree the forks of the branches and the ends of the branches, which we call them key points because key events always happen there, such as choosing a path, getting the help from a worm and arriving at the house he is searching for.

Assume all worms live at key points, and all the branches between two neighboring key points have the same distance of 1. The snail is now at the first fork of the tree.

Our purpose is to find a proper route along which he can find his house as soon as possible, through the analysis of the structure of the tree and the locations of the worms. The only restriction on the route is that he must not go down from a fork until he has reached all the ends grown from this fork.

The house may be left at the end of any branches in an equal probability. We focus on the mathematical expectation of the distance the snail has to cover before arriving his house. We wish the value to be as small as possible.

As illustrated in Figure-1, the snail is at the key point 1 and his house is at a certain point among 2, 4 and 5. A worm lives at point 3, who can tell the snail whether his house is at one of point 4 and 5 or not. Therefore, the snail can choose two strategies. He can go to point 2 first. If he cannot find the house there, he should go back to point 1, and then reaches point 4 (or 5) by point 3. If still not, he has to return point 3, then go to point 5 (or 4), where he will undoubtedly find his house. In this choice, the snail covers distances of 1, 4, 6 corresponding to the circumstances under which the house is located at point 2, 4 (or 5), 5 (or 4) respectively. So the expectation value is (1 + 4 + 6) / 3 = 11 / 3. Obviously, this strategy does not make full use of the information from the worm. If the snail goes to point 3 and gets useful information from the worm first, and then chooses to go back to point 1 then towards point 2, or go to point 4 or 5 to take his chance, the distances he covers will be 2, 3, 4 corresponding to the different locations of the house. In such a strategy, the mathematical expectation will be (2 + 3 + 4) / 3 = 3, and it is the very route along which the snail should search the tree.

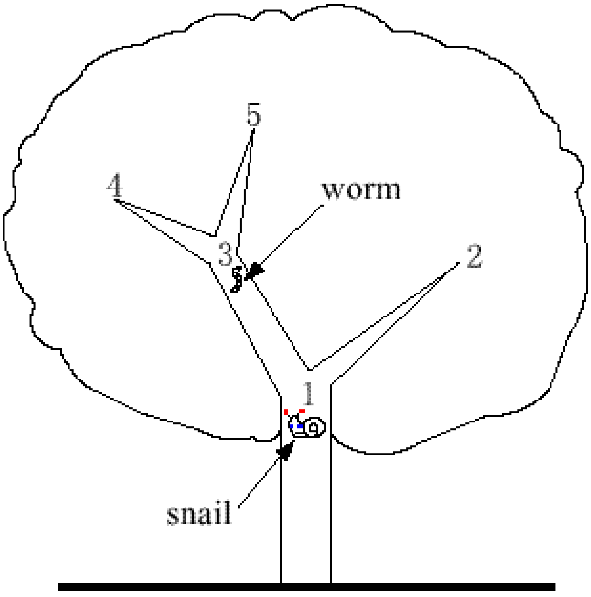

Figure-1

Input

The input contains several sets of test data. Each set begins with a line containing one integer N, no more than 1000, which indicates the number of key points in the tree. Then follow N lines describing the N key points. For convenience, we number all the key points from 1 to N. The key point numbered with 1 is always the first fork of the tree. Other numbers may be any key points in the tree except the first fork. The i-th line in these N lines describes the key point with number i. Each line consists of one integer and one uppercase character 'Y' or 'N' separated by a single space, which represents the number of the previous key point and whether there lives a worm ('Y' means lives and 'N' means not). The previous key point means the neighboring key point in the shortest path between this key point and the key point numbered 1. In the above illustration, the previous key point of point 2 or 3 is point 1, while the previous key point of point 4 or 5 is point 3. This integer is -1 for the key point 1, means it has no previous key point. You can assume a fork has at most eight branches. The first set in the sample input describes the above illustration.

A test case of N = 0 indicates the end of input, and should not be processed.

Output

Output one line for each set of input data. The line contains one float number with exactly four digits after the decimal point, which is the mathematical expectation value.

Sample Input

```
5
-1 N
1 N
1 Y
3 N
3 N
10
-1 N
1 Y
1 N
2 N
```

```
2 N
2 N
3 N
3 Y
8 N
8 N
6
-1 N
1 N
1 Y
1 N
3 N
3 N
0
```

Sample Output

```
3.0000
5.0000
3.5000
```

Beijing 2004-2005