Optimal BST

Problem description

This problem deals with the optimal binary search tree.

For the optimal search tree, we assumed that the search key was always present in the tree. In general this is not true. The search key may not be present in the tree. We can model this situation as follows :

For each i, assume we have probability q_i which gives the probability that a search key, say k, lies between k_i and k_{i+1} (the keys are in increasing order). In addition, let q_0 be the probability that a search key $k < k1$ and q_n be the probability that $k > k_n$. So in addition to the probabilities for a hit, we have probabilities for a miss.

Implement the optimal BST algorithm to find the optimal BST, and the minimum expected value of the number of comparisons.

Input format

The input will consist of three lines. The first line will have an integer n, which denotes the number of keys in the tree. The second and third line will have n values, which denote the probabilities as defined above. Please refer to the pdf document for the exact question format.

Output format

The output format should consist of two lines. The first should tbe the expected value for the number of comparisons in the optimal binary tree. The second line should contain a recursive structure containing the BST itself. It should be written as follows (root(left-BST)(right-BST))

Note 1 : If one of the children of any node is empty, please print () for that child. Please look at the sample I/O for an example of this.

Note 2 : Please print 2 decimal digits whenever you are printing them in your program. The test cases will have 2 decimal digits. You can see this in the sample output.

Note 3 : In case of multiple trees with the same expected cost, submit those with the smallest

elemetn as the root of the tree, and that of the subsequent subtrees.

Note 4 : People for whom the scores were showing as 0, with their code working, the issue was with how SPOJ interpreted spaces. Your code should work now, the judge has been updated. Please do not print any spaces in the output, the first line should have a float value with 2 decimals, and the second line must consist only of parantheses and integers.

Sample input

```
10
0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05
0.01  0.04 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.05
```

Sample output

```
3.70
( 5 ( 3 ( 2 ( 1 ) ( ) ) ( 4 ) ) ( 7 ( 6 ) ( 9 ( 8 ) ( 10 ) ) ) )
```

Scoring method

Please note that apart from the sample test case posted here, there will be other hidden test cases that your code will checked on. The final score you see is a percentage of the test cases you have passed. If you pass only the sample test case, you will get 0/100 (even though your score will be shown as 10/100).

Each of these questions are finally worth the points mentioned in the assignment pdf file. So your credit for this problem shall be your score/100 * points for this question. Your final credit for programming assignment 3 shall be final score / 55 * 100.

Plagiarism and copying

Strict action will be taken against students who are found to be indulging in plagiarism. Please note that changing variable names, removing indentation or moving code around will not help with regards to plagiarism checking.

TA note : The testcases and sample I/O have been updated (13.45, 1/7/17). In case of doubts, please contact the TA at gundeep@iitk.ac.in

The deadline for this problem alone has been extended to 3rd July 2355 hrs. The others need to

be submitted by the original deadline.