TWGSS
OI Team

2023/24 SDIC Practice

Editorial

Daniel Hsieh

5 October 2023

Statistics

	Problem	Attempts	Full	Max	Mean	Std Dev	LQ	Median	UQ	Subtasks	First Solve	
TC24A1	Output the Input	8	100	90	59.375	21.714	40	70	70	100 | 0 90 | 1 70 | 4 55 | 1 25 | 2		
TC24A2	Free Open	7	100	100	33.571	34.301	10	20	70	100 | 1 70 | 1 25 | 1 20 | 1 10 | 2	Username	01:25:10
TC24A3	Out of Sync	4	100	12	3	5.196	0	0	6	100 | 0 12 | 1		
		8	**300**	**202**	**80.222**	**58.579**	**30**	**80**	**117.5**			

Output the Input

TC24A1

Solution

Just do what you are asked!

“It passed the sample, why I get so low mark?”

Common Mistake 1 $\mod M$

You should $\mod M$ every time the value might exceed the range of `int` / `long long`!

e.g. product of three 10^9 is 10^{27}, so you might still overflow!

Of course, please use `long long` if necessary!

Common Mistake 2 Fast I/O

To optimise the I/O speed…

You can use

```
cin.tie(0); cout.tie(0)->sync_with_stdio(false);
cout << '\n';
```

```
verdict: AC
  score: 100.000
runtime: 0.11 s
 memory: 1456 KB
```

Instead of

```
cout << endl;
```

```
verdict: TLE
  score: 85.000
runtime: 1 s
 memory: 1428 KB
```

Free Open

TC24A2

Test Cases 1-5

$1 \leq N, Q \leq 5000$
$1 \leq c_i \leq 4 \times 10^5$

Just simulate!

Score	25

Test Cases 6-7

All doors are initially closed.

Output `0`!

Score	10

Test Cases 8-10

All doors are initially open.

You have to close all the doors one by one.

Therefore, you can use **partial sum** to maintain the sum of cost to close all the doors in the range.

Score	15

Test Cases 11-14

All doors are open or closed with the same probability.

Key Idea: You move right and open door along side, until you reach a suffix of doors in range $[L, R]$ so that all those doors are closed.

Therefore, you can find the range of doors that you need to closed.

As the test data is randomly generated, you can simply loop leftwards from R, and find the first open door x. Then the answer is simply the sum of costs of door $L, L+1, \cdots, x$.

Score	20

Full Solution

To complete the solution, you can pre-compute the first door that is open from each door.

Total Score	100

Out of Sync

TC24A3

Test Cases 1-8

$1 \leq N \leq 5000$

Idea: you need to find the furthest home from house 1 that is remains connected with house 1

Therefore, you may simply use **DFS** to calculate all the distances (after the fibers are cut)!

Time Complexity: $O(N^2 \lg N)$

Score	32

Test Cases 9-11

The tree forms a **chain**, i.e. the i^{th} fiber connects house i and house $i + 1$.

Note that this means you only have to maintain the furthest index of the house that remains connected.

Time Complexity: O(N)

Score	12

Test Cases 12-15

The tree forms a **star**, i.e. the i^{th} fiber connects house 1 and house $i + 1$.

Basically, you can maintain a `multiset` S of distances.

You can precompute the distances once by Dijkstra / DFS, and put them all in S.

Once a fiber is cut, the distance is removed from S. Then you simply output the maximum distance in S every time.

Time Complexity: O(N lgN)

Score	16

Test Cases 16-19

The tree forms a **complete binary tree**, i.e. the i^{th} fiber connects house $i+1$ and house $\left\lfloor \frac{i+1}{2} \right\rfloor$.

Using the same idea from previous special property, you can maintain a set of distances. Once a node is removed, all the distances of relevant nodes are also removed.

Time Complexity: $O(N \lg^2 N)$

Score	16

Full Solution

Using the same idea, notice that each distance (node) is removed once, so if you terminate whenever the node is removed already, the time complexity becomes O(N lgN)!

Total Score	100