South African Computer Olympiad
Final Round 2012
Day 2

Overview

Problem	**visit**	**knockout**	**subway**
Source	visit.java visit.py visit.c visit.cpp visit.pas	knockout.java knockout.py knockout.c knockout.cpp knockout.pas	subway.java subway.py subway.c subway.cpp subway.pas
Input file	stdin	stdin	stdin
Output file	stdout	stdout	stdout
Time limit	.51.5 second	1 second	1 second
Memory limit	64MiB	64MiB	64MiB
Number of tests	10	10	10
Points per test	10	10	10
Detailed feedback	No	No	No
Total points	**100**	**100**	**100**

The maximum total score is 300 points.

South African Computer Olympiad
Final Round 2012
Day 2

Visiting Neighbours

Mark Danoher

Introduction

Carl recently moved into a new neighbourhood only to discover that the poor planning of sidewalks makes visiting neighbours very difficult. He wants to convince the local council to rectify the situation and needs your help to show them how bad it really is.

Task

Carl's neighbourhood consists of a single street of N houses of varying heights. The inexplicably complex system of sidewalks only allows residents to walk between houses that are of exactly the same height, severely limiting their options for social interaction. He has gathered a list h_1 to h_N of the heights of all of the houses on the street and would like you to help him calculate, for each house, how close the nearest house of the same height is. (The distance between house i and house j is $|i - j|$.) In some cases there might not be another house of the same height, leaving the residents of that house with no one else to visit, and Carl would like to know this too.

Example

Suppose there are six houses along the street with heights, in order: 2, 1, 2, 5, 1, 2.

- The residents of houses 1, 3 and 6 can visit one another. House 1 is the closest to house 3 with a distance of 2, and vice versa, but the closest to house 6 is house 3 with a distance of 3.
- The residents of houses 2 and 5 can only visit each other, so they are necessarily the closest to each other with a distance of 3.
- House 4 is the only house of height 5, so its residents cannot visit any other houses.

Input (stdin)

The first line of the input contains a single integer N, the number of houses on the street. The next line contains N space-separated integers h_1 to h_N representing the heights of the N houses.

Sample input

```
6
2 1 2 5 1 2
```

Output (stdout)

Your output should consist of N space-separated integers d_1 to d_N where d_i is the distance from house i to the nearest house of equal height (or -1 if there is no other house of equal height).

Sample output

```
2 3 2 -1 3 3
```

Constraints

- $1 \leq N \leq 100\,000$
- $1 \leq h_i \leq 100\,000$

Additionally, in 50% of the test cases:

- $1 \leq N \leq 1\,000$

Time limit

.51.5 second.

Scoring

A correct solution will score 100% while an incorrect solution will score 0%.

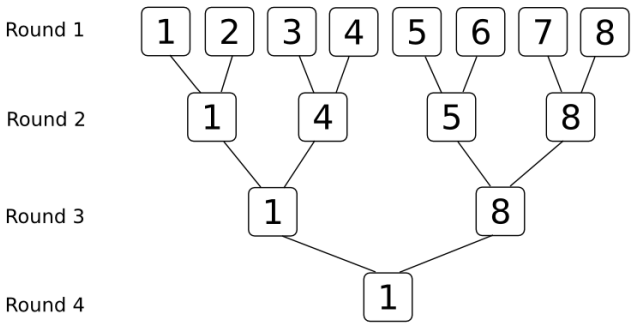
Knockout Tournament

Kosie van der Merwe

Introduction

Recently a knockout tournament was held between Bruce's and Carl's messenger swallows, to see which swallow could deliver their coconut the most quickly.

In this knockout tournament pairs of swallows compete and the loser is removed from the competition. The winner moves on to the next round. Below is an example of a knockout tournament with $N = 8$ (note swallows are numbered from 1 to N):



In round 1 swallow 1 and 2 compete and 1 emerges victorious. Similarly, 3 and 4 competed and 4 won. In round 2, 1 and 4 compete and 1 wins.

After the tournament you wish to know the relative ranking of all the swallows, the problem is not all pairs of swallows competed. It is assumed if swallow A beats swallow B and swallow B beats swallow C, then swallow A would beat swallow C. That is winning is transitive.

In the example tournament we know swallow 1 is first, but swallow 2 can be anywhere from second (having lost to the eventual winner) to last.

Task

Given the number, N, of swallows (which is guaranteed to be a power of two) and the tournament chart you have to find the lowest and highest rank each swallow could have achieved.

Input (stdin)

The first line of the input contains a single integer, N. The next $\log_2(N)+1$ lines contain the contestants for each round as a list of space separated integers. ($\log_2(N) = k$ where $N = 2^k$)

The second line contains contestants for round 1 (a total of N contestants), the third line contains the contestants for round 2 and so on.

It is guaranteed the contest chart will be valid, that is:

- All contestants will be numbered from 1 to N.
- No contestant will participate more than once in a round.
- If round i has 2^k contestants then round $i + 1$ will have 2^{k-1} contestants.
- Contestant number i (numbered from left to right from 0) in round $j + 1$ will be the same as either contestant $2i$ or $2i + 1$ in round j. Essentially, the contestants in subsequent rounds will be victors from previous rounds

Note for convenience the swallows will be ordered by their number in the first round.

Sample input

```
8
1 2 3 4 5 6 7 8
1 4 5 8
1 8
1
```

Output (stdout)

The output consists of N lines of pairs of space separated integers. On the line ith output the highest and lowest rank swallow i could have achieved, respectively.

Sample output

```
1 1
2 8
3 8
2 7
3 7
4 8
3 8
2 5
```

Constraints

- N is a power of 2.
- $1 \leq N \leq 131\,072$.

Additionally, in 50% of the test cases:

- $N \leq 8\,192$.

Time limit

1 second.

Scoring

A correct solution will score 100% while an incorrect solution will score 0%.

Subway equivalence

Gwylim Ashley

Introduction

Rob the Robot has been programmed to map the subway networks of various cities automatically. For each city, he has explored the subway network and recorded some information about his exploration, which will be used to create a map. Unfortunately, due to a programming error, Rob has explored some of the subway networks more than once, and he does not know how many cities he has visited in total.

Given the recorded information for each subway network, you must help determine the number of different subway networks that Rob has explored.

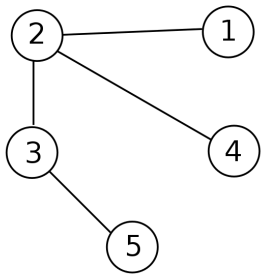
Task

A subway network consists of a number of *stations*, and a number of *lines*, each of which connects two stations, such that there is exactly one way to travel between any two stations without traveling on the same line twice (in other words, there are no loops in the network). Rob explores each subway network by starting at a random station, and then traveling along the lines to visit each other station, before returning to his starting position. He does this in the least possible number of trips, so that he travels along each line exactly twice: once going away from his starting position, and then once returning towards it. However, he does not attempt to visit the stations in any particular order.

When Rob travels along a line for the first time, he writes down a 0, and when he travels along it for the second time, he writes down a 1, hence obtaining a string of 0s and 1s describing the subway network. Each city has a unique subway network, so it is impossible to obtain the same strings for two different cities, regardless of the station that Rob starts at. However, it may be possible to obtain different strings for the same network.

He now has a collection of strings of 0s and 1s, each of the same length V, describing different subway networks; however, some of them may in fact describe the same subway network. Your goal is to determine, given this information, the number of different subway networks that he has explored.

Example



In the above example, if Rob starts at station 4, and takes the route $4 \to 2 \to 1 \to 2 \to 3 \to 5 \to 3 \to 2 \to 4$, then he will obtain the string 00100111 describing the subway network. He could also obtain the string 00011011 by starting at station 4, and taking the route $4 \to 2 \to 3 \to 5 \to 3 \to 2 \to 1 \to 2 \to 4$. However, he could never obtain the string 00001111 no matter what station he starts at, or what route he takes. This string then must correspond to a different subway network.

Input (stdin)

The first line of input will consist of an integer N, the number of records Rob has. The following N lines will each consist of a string of 0s and 1s. All strings will be of the same length V.

Sample input

```
3
00100111
00011011
00001111
```

Output (stdout)

The output should contain a single integer M, the number of distinct subway networks that Rob has explored.

Sample output

```
2
```

Constraints

- $1 \leq N \leq 1000$
- $1 \leq V \leq 30$

Additionally, in 30% of the test cases:

- $1 \leq N \leq 50$
- Whenever Rob visits the same city more than once, he starts at the same station in each case.

Additionally, in 60% of the test cases:

- Whenever Rob visits the same city more than once, he starts at the same station in each case.

Time limit

1 second.

Scoring

A correct solution will score 100% while an incorrect solution will score 0%.