General Information and Rules

The theoretical part of the second round takes place at ETH Zurich on February 28, 2015. The participants have to solve four tasks in four hours. The solutions must be written on paper. For each task begin a new sheet of paper. The participants are not allowed to use any literature, electronik devices (except simple watches) or other aiding materials.

The solutions will be graded according to similar criteria as the first theoretical round. The most important criteria are correctness and asymptotic run time. The quality of the description and the arguments asserting the correctness will also be taken into account.

Each solution should contain:

- **Description of the solution**. This part should explain the used algorithm, analyse its asymptotic runtime and space usage and explain why the approach is correct. The description should be understandable without looking at its source code.

- **Program**. This part should contain the source code of the most important parts of the algorithm in Pascal, C or C++. You can skip simple parts like Input, Output, simple mathematical expressions, ...

Round 2T Task descriptions

2T–1: Gossip

Mouse Mark runs a social network called Mousebook. As operator, he knows the complete structure of the network: I.e., he knows who is friends with whom (and with whom he is not). Information spreads in the network as follows: Each Mouse who receives a new message first reads it carefully, and forwards it to all its friends immediately afterwards. Messages on Mousebook reach the receiver in the moment they are sent. Each Mouse on Mousebook has a fixed reading speed, given in (whole) seconds per letter (not counting spaces). From earlier interactions on Mousebook, Mouse Mark was able to determine this speed for each Mouse exactly.

The communication is initiated by Mouse X, who invents a new piece of gossip and sends it to all its friends.

Mouse Mark is interested in how fast information can spread in his network. I.e., he wants to predict how long it takes after the initial sending, until all Mice on Mousebook have read the new piece of gossip. Here, we assume that this can always happen in finite time, and that at each moment there is only one piece of gossip circulating. Furthermore, each Mouse reads and forwards some piece of gossip only once; if it receives a message another time, it just ignores it.

Task Mousebook has n Mice members. For each Mouse, we are given its reading speed as well as a list of its friends. Friendship is always mutual. Furthermore, we are given Mouse X, together with the piece of gossip that it has invented. After how many seconds has the gossip been read by all Mice in the network?

For the first two subtasks, there is a concrete example, for which you should determine the solution; for each subtask you should furthermore find and describe an algorithm that can solve arbitrary examples.

Note, that like for all other tasks of 2T, you can largely omit the implementation of input/output. This includes, for example, the computation of the length of the gossip, or the conversion of the friendship lists in a more handy format. You still have to precisely specify which variable contains which input value though.

a) (30 points) For this subtask, we assume that all Mice have identical reading speed, and that this speed is 1 second per letter.

Concrete Example (10 points):

Mouse	**s/l.**	**Friends**
Anna	1	Brigitte, Joachim, Stofl
Brigitte	1	Anna, Horst, Stofl
Charlie	1	Miriam, Stofl
Horst	1	Brigitte, Jessica, Josef
Jessica	1	Horst
Joachim	1	Anna, Josef, Richard
Josef	1	Horst, Joachim, Sophia
Miriam	1	Charlie, Sophia
Monika	1	Richard, Stofl
Richard	1	Joachim, Monika
Sophia	1	Josef, Miriam
Stofl	1	Anna, Brigitte, Charlie, Monika

Mouse X: Monika
Gossip: “The moon is made of Swiss cheese” (26 letters)

For the complete description of the algorithm for this subtask (including explanation/proof of correctness/cost analysis), you can obtain 20 points.

b) (50 points) For this subtask, there are no further constraints on the reading speeds.

Concrete Example (10 points):

(The network is identical to that of the last example, the reading speeds are different.)

Mouse	s/l.	Friends
Anna	1	Brigitte, Joachim, Stofl
Brigitte	9	Anna, Horst, Stofl
Charlie	2	Miriam, Stofl
Horst	1	Brigitte, Jessica, Josef
Jessica	3	Horst
Joachim	7	Anna, Josef, Richard
Josef	2	Horst, Joachim, Sophia
Miriam	1	Charlie, Sophia
Monika	3	Richard, Stofl
Richard	5	Joachim, Monika
Sophia	3	Josef, Miriam
Stofl	1	Anna, Brigitte, Charlie, Monika

Mouse X: Jessica
Gossip: "Cheese is healthy" (15 letters)

For the complete description of all components of the algorithm for this subtask (including explanation/proof of correctness/cost analysis) you can obtain 40 Points.

c) (20 points) Recently, a new Mouse Y has signed up to the network. Mouse Y is just as creative as Mouse X in inventing new gossip. It now frequently happens, that two different pieces of gossip are circulating at the same time, one which was written by Mouse X and one which was written by Mouse Y. Therefore, it may now also happen, that some Mouse is already busy reading one message, when a new message arrives already. In this case, the Mouse starts reading the second message as soon it has forwarded the first message. If two different messages arrive at exactly the same time, then the message written by Mouse Y is read first, sind Mouse X has been discredited for occasionally sending around irrelevant false reports. Here, we assume that Mouse X and Mouse Y send their gossip at the same time, and that at each moment, only those two messages are circulating. Explain, which modifications are to be made to algorithm/proof of correctness/cost analysis in order to be able to determine, how long it will take until both messages have been read by all Mice.

2T–2: Washing dishes

Mouse Stofl is working as a dishwasher in a noble restaourant. Each evening, he is given a stack of dirty dishes, which he must wash in the lavatory before putting them into the cupboard. Thereby he has to consider one peculiarity of the restaurant: All dishes are of different sizes, and he must organize them as a stack sorted by diameter (largest at the bottom, smallest on top) inside the cupboard, since otherwise the waiters would confuse themselves. Furthermore, there is not enough space in the lavatory; the dishes cannot be placed beside each other and have to be organized as a stack there as well. Therefore, we will assume that Stofl can perform the following two steps exclusively:

1. He takes the topmost dish from the stack of dirty dishes and puts it on top of the stack in the lavatory. Since the lavatory is filled with a powerful dish cleanser, the dish is immediately cleaned without Stofl's additional involvement.

2. He takes the dish that was put into the lavatory last and puts it on top of the stack in the cupboard.

(Other conceivable steps, like exchanging dishes within the stack, or moving multiple dishes at once are not possible, since Stofl requires both arms to carry a dish – as a Mouse, he is not very powerful.)

Stofl combines those steps in an arbitrary sequence in order to complete his task (as he is not immortal, there can only be a finite number of steps). Even though he is a very experienced dishwasher, he sometimes cannot do it; he even suspects that it is impossible, no matter how cleverly he acts, in case the dishes are sorted in an unsuitable order on the stack. Can you help Stofl by writing a program, which will tell him immediately in case he is working with such a stack?

Task Let n be the number of dishes on the "dirty" stack. Your program is given a list of n numbers as the input, which denote the diameter of the dishes. Then it should decide whether this stack can be put into the cupboard sorted by a finite sequence of steps like the above.

Input The first line of the input contains the number n of dirty dishes. The next line contains n numbers, the diameters of the dishes in ml (mouse lengths). For $1 \leq i \leq n$ the i-th number denotes the diameter of the i-th dish, starting from the top of the stack. You may assume that the diameters are pairwise distinct.

Output In case Stofl is able to put the dishes into the cupboard sorted, your program should print "YES", and "NO" otherwise. Note that it is very important to argue why your program is correct.

EXAMPLE:

Input:	**Output:**
7 2.8 4.1 3.9 5.2 7.6 1.1 6.9	NO
Input:	**Output:**
8 0.01 0.15 200 1 2 70.2 3.5 4	YES

2T–3: Hidden Cheese

Mouse Stofl lives together with many other mice in a big mouse hole. The hole also contains a hidden chamber where they store the stock of their best cheese. The chamber contains an entrance and a labyrinth of paths. Paths are occasionally split into two paths. There is exactly one path leading to the cheese stock. In order to ensure that the mice are able to find the cheese, they have put signs at each junction, which are supposed to lead to the cheese. Unfortunately, the chamber was recently renovated, and the construction workers have accidentally changed all signs to point to the left. As this embarrassed the construction company, they didn't tell the mice; they still think that the signs lead them to the cheese stock.

Each night, one of the mice has night duty and has to guard the cheese, but instead decides to go eat some delicious cheese. Each mouse follows the signs, but inverts the direction of all signs on its path, in order to divert the other mice and have the cheese for itself.

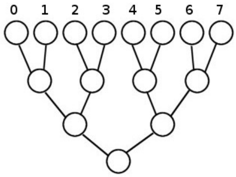
Will Stofl find the cheese?

Task The entrance of the chamber is given by a full binary tree of height h. This means, that Stofl starts at one junction (this is the entrance) and at each junction he can only decide to turn left or right. Furthermore, he cannot move further after having passed exactly $h-1$ junctions. The leaves (i.e. the ends of the possible paths) are numbered from 0 to $2^{h-1}-1$, where 0 is the leaf reached when always going to the left.
Stofl has to guard the cheese in the n-th night. Note that Stofl is a C programmer and hence starts counting the nights at 0.

a) 10 points For this subtask, let $h = 4$ and $n = 5$. Where does the cheese need to be stored so that Stofl will get it?

b) 80 points For this subtask, let $n < 2^{h-1}$. Write an algorithm that determines at which leaf Stofl arrives. As always, it is important to deliver a

Figure 1: Complete binary tree with $h = 4$



complete description (including runtime analysis/argument for correctness).

c) 10 points Let n be arbitrary (i.e. it can be larger or equal to 2^{h-1}). Which modifications have to be made?

2T–4: Boarding School

Mouse Rowan is headmaster of a prestigeous English boarding school. Since some time, he is bothered by severe violations of the library rules: Some students take books from the library without following the prescribed lending procedures. Even though each book was eventually put back, the official order of the books was not preserved.

Luckily, one of the evil students was caught in the act. In order to prevent further violations of the school's rules by him and others, Rowan will punish the student. The severity of the punishment shall be proportional to the number of illegally taken books. Rowan wants to show mercy: In case this number cannot be determined precisely, he wants to assume the lowest possible number. This means, that he wants to be able to prove conclusively that at least so many books where taken (and then put back). He will need your help.

Task There are n books in the library. Initially, they are sorted, and for simplicity's sake, we number them from 1 to n. The evil student repeatedly takes a book and puts it back at another position. This way, the initial sequence

`12345` $(n = 5)$

might be turned in one step (take book 4, put it back in front of book 1) into the sequence

`41235`

and in another step (take book 3, put it back between book 1 and book 2)

`41325.`

The task can hence be described as follows: Given n and a sequence (permutation, in case you are familiar with this terminology) of n books, how many steps are required in order to turn the initial sequence (the identity permutation) into the given one? Each of the following subtasks contains a concrete example, which you should compute; furthermore you should find an algorithm, which can solve arbitary examples (given the assumptions of the subtask).

Note that the first two subtasks make additional assumptions where a book may be put back. For example, the second step of the above example is not valid given those assumptions.

Input The first line of the input contains an integer $n \geq 1$, the number of books. The next line contains the integers from 1 to n, separated by single spaces, in an arbitrary order; this is the order of the books.

Output The output should consist of a single number, the minimum number of steps necessary to turn the initial sequence into the sequence given in the input.

a) (30 points) For this subtaks we assume that a taken book can only be put back at the beginning of the sequence. The first step of the above example would hence be valid, but not the second one.

Concrete Example (5 Punkte):

```
7
6231457
```

For the complete description of the algorithm for this subtask (including explanation/proof of correctness/cost analysis), you can obtain 25 points.

b) (35 points) For this subtask, we assume that a taken book can be put back either at the beginning or at the end of the sequence. Again, the first step of the above example would be valid, but not the second one. However, `41235` $\rightarrow$ `41352` would be a valid second step (take book 2 and put it back at the end).

Concrete Example (5 Punkte):

```
7
2534716
```

For the complete description of the algorithm for this subtask (including explanation/proof of correctness/cost analysis), you can obtain 30 points.

c) (35 points) For this subtask, there are no further constraints on the number of allowed steps. In the above example, both steps would be valid.
Concrete Example (5 Punkte):

```
7
3126574
```

For the complete description of the algorithm for this subtask (including explanation/proof of correctness/cost analysis), you can obtain 30 points.