

2019 Canadian Computing Olympiad

Day 1, Problem 1

Human Error

Time Limit: 1 second

Problem Description

Justin and Donald are playing their favourite game: hop chess. You probably haven't heard of it, but the rules are pretty simple.

The board is a rectangular grid, with each square of the board initially having exactly one player's piece on it. Justin's pieces are denoted as J , with Donald's being D . Each player has at least one piece on the grid initially.

The game begins with Justin playing first. On a turn, a player may move one of his pieces to capture (and thus remove from the board) a neighbouring piece (not necessarily the opponent's). A piece X is said to be neighbouring Y if it is either up, down, left, or right of Y . If such a move cannot be made, then the active player loses.

In an ideal world, both Justin and Donald are perfect logicians, and capable of discerning an optimal strategy for any board. Then perhaps we might be interested in who of the two would win. But that wouldn't be very realistic. Indeed, when playing, Justin and Donald can both come up with a relatively good solution; exactly how good it tends to be is determined by their error factors, J and D respectively.

Formally, the active player with error factor A first chooses a *proposal set*: either the set of all possible moves if there are A or less possible moves or some subset of size A from the set of possible moves if he has more than A possible moves. Then, from this *proposal set*, the player selects a move randomly with equal probability.

Both players, when given the choice of choosing their *proposal set*, chooses the most optimal such set (one which produces the highest probability of winning), knowing that the other player always chooses their *proposal set* optimally as well.

The natural question is then: exactly what is the probability that Justin wins a game of hop chess, given the initial board, J , and D ?

Input Specification

Input will begin with two space-separated positive integers R, C ($R \cdot C \leq 13$). On the next R lines will be strings of C characters drawn from the set $\{J, D\}$, describing the initial board state. Finally, there will be two space-separated integers, J, D ($1 \leq J, D \leq 13$)

Output Specification

Output a single floating point number rounded to 3 decimal places: the probability that Justin wins.

Sample Input 1

```
1 3
JJD
3 1
```

Output for Sample Input 1

```
0.667
```

Explanation of Output for Sample Input 1

Note that Justin has 3 possible moves (note that `_` indicates an empty cell in all explanations below):

- he moves his first piece right, capturing his second piece, and ensuring his loss by having the board appear as `_JD`;
- he moves his second piece right, capturing Donald's piece and securing victory, with the board as `J_J`;
- or he moves his second piece left, capturing his own piece, but leaving Donald unable to move, thus also winning, with the board as `J_D`.

Clearly the latter 2 cases are optimal—but since Justin has error factor 3, there is a $1/3$ chance that he chooses the option causing him to lose. Thus he wins with probability $2/3$.

Sample Input 2

```
2 2
JJ
DD
3 1
```

Output for Sample Input 2

```
0.000
```

Explanation of Output for Sample Input 2

There is no hope for Justin to win.

To see why, notice that Justin has 4 possible first moves:

<code>J_</code>	<code>_J</code>	<code>J_</code>	<code>_J</code>
<code>DD</code>	<code>DD</code>	<code>DJ</code>	<code>JD</code>

He can pick any subset of size three from the above moves.

However, Donald will always pick his most optimal move. Regardless of Justin's first move, Donald will leave the board in one of the following configurations:

D_ _J _D J_
D D D_ _D

all of which will cause Justin to lose.

2019 Canadian Computing Olympiad

Day 1, Problem 2

Sirtet

Time Limit: 2 seconds

Problem Description

In a fancy new zero-person video game, Sirtet, the game is a rectangular grid with N rows and M columns. Before the game begins, some grid cells are blank (denoted as `.`) and others are filled (denoted as `#`). The filled squares represent a set of objects, and the filled squares that are adjacent (horizontally or vertically) should be considered to be part of the same rigid object. For example, this initial grid:

```
. . # .  
# # . #  
. # # .  
# . . .  
# . . .
```

has four objects, shown below:

```
##      #      #      #  
##      #
```

When the game begins, the objects fall straight down the grid, all at the same speed. Each object continues to fall straight down until it either touches the bottom row, or has some part of it land directly on top of another object, at which point it stops. What will be the final state of the grid?

Input Specification

The first line contains two space-separated positive integers N and M ($N \cdot M \leq 10^6$).

The following N lines contain M characters each, describing the initial state of the grid. If the j -th column of the i -th row of the grid contains a block, the corresponding character in the input will be a `#`, otherwise it will be a `.` character.

For 10 of the 25 available marks, $N \cdot M \leq 2000$.

For an additional 6 of the 25 available marks, $M = 2$.

Output Specification

Output N lines contain M characters each, describing the final state of the grid. If the j -th column

of the i -th row of the grid contains a block, the corresponding character in the input will be a #, otherwise it will be a . character.

Sample Input

```
5 4
. . # .
# # . #
. # # .
# . . .
# . . .
```

Output for Sample Input

```
. . . .
. . . .
### .
### .
# . . #
```

2019 Canadian Computing Olympiad

Day 1, Problem 3

Winter Driving

Time Limit: 1 second

Problem Description

In the Great White North, there are N cities numbered from 1 to N . There are A_i citizens living in city i . There are $N - 1$ roads numbered from 2 to N . Road j connects city j and city P_j , where $P_j < j$. There are at most 36 roads connected to any city.

During winter, all roads will be converted into one-way highways due to dangerous driving conditions. That is, road j will become a highway that is either one-way from city j to city P_j or one-way from city P_j to city j .

Every citizen wants to send a holiday card to every other citizen. Citizen x can send a card to citizen y if it is possible to travel from the city x lives in to the city y lives in using only highways.

What is the maximum number of holiday cards that can be sent after converting all roads to highways?

Input Specification

The first line contains one integer N ($2 \leq N \leq 200\,000$).

The second line contains N integers A_1, \dots, A_N ($1 \leq A_i \leq 10\,000$).

The third line contains $N - 1$ integers P_2, \dots, P_N ($1 \leq P_j \leq j$).

Let D be the maximum number of roads connected to any city. It is guaranteed that $D \leq 36$.

For 5 of the 25 available marks, $N \leq 10$.

For an additional 5 of 25 available marks, $N \leq 1\,000$ and $D \leq 10$.

For an additional 5 of 25 available marks, $D \leq 18$.

For an additional 5 of 25 available marks, there will be 37 cities, where one city is connected to 36 other cities, and these other 36 cities are only connected to this one city.

Output Specification

Print one line with one integer, the maximum number of cards that can be sent after converting all roads to highways.

Sample Input

4

3 3 4 1

1 2 1

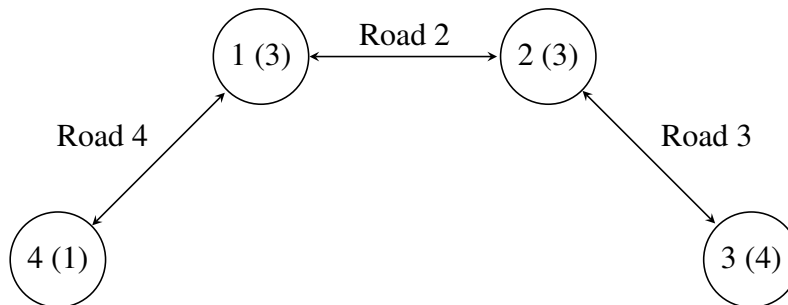
Output for Sample Input

67

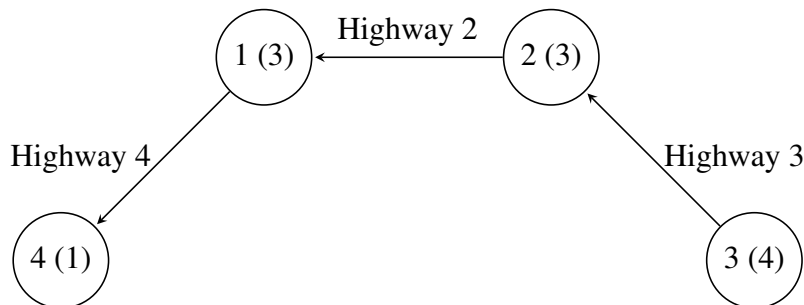
Explanation of Output for Sample Input

One possible way of converting roads to highways is for road 2 to become one-way from city 2 to city 1, road 3 to become one-way from city 3 to city 2, and road 4 to become one-way from city 1 to city 4.

Consider the following pictures, with the cities and associated population (in parentheses) for the initial roads



and what it looks like after all roads are converted to highways:



Every citizen in city 3 can send 3 holiday cards to city 3 citizens, 3 holiday cards to city 2 citizens, 3 holiday cards to city 1 citizens, and 1 holiday card to the city 4 citizen, for a total of 40 holiday cards sent out of city 3.

Similarly,

- city 2 citizens send 6 holiday cards each, for a total of 18 holiday cards.
- city 1 citizens send 3 holiday cards each, for a total of 9 holiday cards.
- the city 4 citizen cannot send any holiday cards.

A total of $40 + 18 + 9 = 67$ holiday cards are sent.

2019 Canadian Computing Olympiad

Day 2, Problem 1

Card Scoring

Time Limit: 6 seconds

Problem Description

You have a deck of n cards. Each card has a value: the card values lie between 1 to n , possibly with some repeated values, and possibly with some values never occurring. There is also a special integer k that will be used in calculating a score.

You are playing a game that involves drawing all the cards from the deck one by one. When you draw a card, you may choose to either add it to your hand or discard it. You may also *score* your entire hand at any time. When you score a hand with x cards, you gain $x^{\frac{1}{2}k}$ points and then you discard all cards in that hand. At any point in time, your hand may only contain cards with the same number on them. Given the initial order of the cards in the deck, what is the maximum possible score you can obtain?

Input Specification

The first line of input will contain two space-separated integers k and n . The value k is the value used in the expression $x^{\frac{1}{2}k}$ to compute points ($2 \leq k \leq 4$). The value n is the number of cards in the deck ($1 \leq n \leq 1\,000\,000$). The next n lines contain one integer per line, where the i th of these n lines is the i^{th} card from the top of the deck ($1 \leq i \leq n$).

For 4 of the 25 marks available, $1 \leq n \leq 20$.

For an additional 1 of the 25 marks available, $1 \leq n \leq 300$, $k = 2$.

For an additional 5 of the 25 marks available, $1 \leq n \leq 300$.

For an additional 3 of the 25 marks available, $1 \leq n \leq 5\,000$.

For an additional 3 of the 25 marks available, $k = 4$.

Output Specification

Output one floating point number, which is the maximum score you can obtain from playing optimally.

If your answer is p and the correct answer is q , then your answer will be considered correct if

$$\frac{|p - q|}{q} \leq 10^{-6}.$$

Sample Input 1

3 5
1
2
2
1
1

Output for Sample Input 1

6.656854249

Explanation for Output for Sample Input 1

We know the cards we draw in order are $[1, 2, 2, 1, 1]$ and that discarding a hand of x cards gives us a score of $x^{1.5}$.

The optimal strategy is to draw one card, score that hand, draw two cards, score that hand, and draw two more cards and score that hand. This strategy gives a score of $1^{1.5} + 2^{1.5} + 2^{1.5} \approx 6.656854249$.

Sample Input 2

4 5
1
2
2
1
1

Output for Sample Input 2

9.0

Explanation of Output for Sample Input 2

We know the cards we draw in order are $[1, 2, 2, 1, 1]$ and that scoring a hand of x cards gives us a score of x^2 .

An optimal strategy is to take all cards with 1, and score them all at the end. This strategy gives a score of $3^2 = 9$. Note that taking the first card and scoring, then taking the next two cards and scoring, and then taking the final two cards and scoring, will also yield $1^2 + 2^2 + 2^2 = 9$.

2019 Canadian Computing Olympiad
Day 2, Problem 2
Marshmallow Molecules

Time Limit: 4 seconds

Problem Description

Hannah is building a structure made of marshmallows and skewers for her chemistry class. The structure will contain N marshmallows, numbered from 1 to N . Some marshmallows will be connected by skewers. Each skewer connects two marshmallows.

Hannah has M requirements for her structure. Each requirement is given as a pair (a_i, b_i) , which means that there must be a skewer connecting marshmallow a_i and marshmallow b_i .

To ensure the stability of the structure, the following requirement must also be satisfied: if $a < b < c$, and if there is a skewer connecting marshmallow a to marshmallow b , and if there is a skewer connecting marshmallow a to marshmallow c , then there must also be a skewer connecting marshmallow b to marshmallow c .

Due to austerity measures imposed by the principal's office, skewers are scarce in Hannah's school. Find the minimum number of skewers necessary to satisfy all requirements.

Input Specification

The first line contains two space-separated integers N and M ($1 \leq N, M \leq 10^5$).

The next M lines each contain two space-separated integers, with the i -th line containing a_i and b_i ($1 \leq a_i < b_i \leq N$). All M pairs (a_i, b_i) are distinct.

For 5 of the 25 marks available, $N \leq 100$.

For an additional 5 of the 25 marks available, $N \leq 5000$.

For an additional 5 of the 25 marks available, for all $1 \leq j \leq N$, there is at most one pair (a_i, b_i) such that $b_i = j$.

Output Specification

Output a single integer, the minimum total number of skewers.

Sample Input 1

6 4
1 2
1 4
4 6
4 5

Output for Sample Input 1

6

Explanation for Output for Sample Input 1

In addition to those already required, there must be skewers between the pairs of marshmallows (2, 4) and (5, 6).

Sample Input 2

7 6
2 3
2 6
2 7
1 3
1 4
1 5

Output for Sample Input 2

16

2019 Canadian Computing Olympiad
Day 2, Problem 3
Bad Codes

Time Limit: 1 second

Problem Description

Your friend has constructed a code that they want to use to send secret messages to you. The messages will only be composed of N different symbols and each symbol will correspond to one binary sequence with at most M bits.

However, you are not sure the code is going to work: there is a chance that a binary sequence can correspond to two (or more) different messages.

For example, if the code was:

$A \rightarrow 101$
 $B \rightarrow 10$
 $C \rightarrow 1$
 $D \rightarrow 100$

then the binary sequence 101 could be correspond to either A or BC.

Your job is determine the length of the shortest binary sequence that corresponds to two different messages, or determine that there are no binary sequences which correspond to two different messages.

Input Specification

The first line of input will contain two space-separated integers N and M ($1 \leq N, M \leq 50$). The next N lines of input each will have at least one and at most M characters from the set $\{0, 1\}$.

For 4 of the 25 marks available, $N = 4$ and $M \leq 6$.

For an additional 7 of the 25 marks available, each of the binary sequences contains exactly one 1 bit. For example, the sequences 00100 or 1000 would be valid in this case.

Output Specification

Output will be one line long.

If there is a binary sequence that corresponds to two (or more) messages, print the length of the shortest such binary sequence; otherwise, output one line containing -1 .

Sample Input 1

4 3
101
10
1
100

Output for Sample Input 1

3

Explanation of Output for Sample Input 1

This is the sample in the problem description.

Sample Input 2

4 4
1011
1000
1111
1001

Output for Sample Input 2

-1

Explanation of Output for Sample Input 2

There is no binary sequence that corresponds to more than one message. Notice that since each code is 4 bits, and none are the same, every encoding of $4k$ bits can be uniquely decoded into k characters.