Fall 2020
Problem Solving through Computational Thinking ECE30017

Week 4

- C3. Two Trains
  Deadline: 11:59 PM, 25 Septeber (Fri)
- P4. Tower of Hanoi
  Deadline: 4:00 PM, 28 Septeber (Mon)

C3. Two Trains

There are *N* cities identified by 1, 2, 3, ..., *N*. Some of two cities are connected directly with railroad.

A transportation corporation has two trains called Alpha and Beta: Train Alpha takes the *p* amount of fuel for moving between two adjacent cities, whereas trains Beta takes the *q* amount of fuel. Interestingly, the two trains can be united. The united train takes *r* amount of fuel for $r < p + q$. Assume that the fuel consumption of a train is the same for transiting any two adjacent cities.

Suppose that Alpha is located at City 1, and Beta is at City 2. And both trains are heading to the same destination City N.

Write a program that finds the minimum amount of total fuel consumption for moving the two trains to City N. Note that the two trains may be united at a city to save fuels for the rest of the travel. Assume that it takes no fuel to unite them.

Requirements

Input

The input data is given to the standard input. The first line has five integers *p*, *q*, *r*, *N*, and M, for each of these numbers is no more than 40000. *N* indicates that there are N cities identifies by 1, 2, ..., *N*. The 2nd to (*M*+1)-th lines contain *M* pairs of identifiers of two cities connected directly with a railroad.

Output

Print out the minimum amount of the total fuel needed for the two trains within 0.5 seconds

Example

input file

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

output file

```
22
```

Team for C3

Team 301	강하영	정현섭	
Team 302	김기훈	홍원표	
Team 303	김석진	전해주	
Team 304	김승우	임예찬	
Team 305	김준서	윤다은	
Team 306	김유진	박수현	
Team 307	김윤정	최우석	
Team 308	김지원	윤지영	
Team 309	박민준	최재혁	
Team 310	송수근	정진혁	
Team 311	송진범	신희주	
Team 312	이예준	한찬솔	
Team 313	지성민	황소정	
Team 314	정희석	홍석현	
Team 315	정원식	정예은	한정섭

PA 4. Tower of Hanoi

The Tower of Hanoi is a game to play with 3 rods (rod 1, rod 2, and rod 3) and n disks of different sizes (disk 1 disk 2, …, disk n). Disk i is lager than disk j if only if $i > j$. The game starts with the disks distributed upon three rods. A disk can be stacked upon another disk in a rod if the former is smaller than the latter. Thus, the disks in each rod are stacked in the ascending order of their sizes, that is, the top-most one is the smallest one in the rod.

The goal of the game is to place all disks in a certain rod (calling it "destination" rod) with the minimum number of disk moves. The game player is allowed to move one disk at a time, from rod to rod as long as the above-stated condition is satisfied.

Note that this game is a generalized version of the traditional Tower of Hanoi game which starts with all disks stacked in one specific rod. It is known that, in the traditional game, the minimum num (ber of disk moves is 2^n -1 for n disks.

Write a program that reads the initial state of the game and then finds the minimum number of disk moves to finish the game. (continued)

Requirements

Input data

- The first line from the standard input has two numbers n and k. n stands for the number of disks where $1 \leq n \leq 50$. k represents the destination rod where $1 \leq k \leq 3$.
- The second to forth lines represent how initially the disks are stacked over three rods. Each line starts with n_i, the number of the disks in rod $i, 1 \leq i \leq 3$ followed by n_i integers that represent the disks initially stacked from bottom to top in rod i.

Output

- write the number of the minimum disks moves to the standard output
- your program must generate the result within 0.5 second.

Examples of test cases

Input 1

```
4 2
2 4 3
1 2
1 1
```

Output 1

```
13
```

Input 2

```
4 3
4 4 3 2 1
0
0
```

Output 2

```
15
```