

## E. Tourism

time limit per test: 2 seconds  
 memory limit per test: 256 megabytes  
 input: standard input  
 output: standard output

Alex decided to go on a touristic trip over the country.

For simplicity let's assume that the country has  $n$  cities and  $m$  bidirectional roads connecting them. Alex lives in city  $s$  and initially located in it. To compare different cities Alex assigned each city a score  $w_i$  which is as high as interesting city seems to Alex.

Alex believes that his trip will be interesting only if he will not use any road twice in a row. That is if Alex came to city  $v$  from city  $u$ , he may choose as the next city in the trip any city connected with  $v$  by the road, except for the city  $u$ .

Your task is to help Alex plan his city in a way that maximizes total score over all cities he visited. Note that for each city its score is counted at most once, even if Alex been there several times during his trip.

### Input

First line of input contains two integers  $n$  and  $m$ , ( $1 \leq n \leq 2 \cdot 10^5$ ,  $0 \leq m \leq 2 \cdot 10^5$ ) which are numbers of cities and roads in the country.

Second line contains  $n$  integers  $w_1, w_2, \dots, w_n$  ( $0 \leq w_i \leq 10^9$ ) which are scores of all cities.

The following  $m$  lines contain description of the roads. Each of these  $m$  lines contains two integers  $u$  and  $v$  ( $1 \leq u, v \leq n$ ) which are cities connected by this road.

It is guaranteed that there is at most one direct road between any two cities, no city is connected to itself by the road and, finally, it is possible to go from any city to any other one using only roads.

The last line contains single integer  $s$  ( $1 \leq s \leq n$ ), which is the number of the initial city.

### Output

Output single integer which is the maximum possible sum of scores of visited cities.

### Examples

input	Copy
5 7 2 2 8 6 9 1 2 1 3 2 4 3 2 4 5 2 5 1 5 2	
output	Copy
27	

input	Copy
10 12 1 7 1 9 3 3 6 30 1 10 1 2 1 3 3 5 5 7 2 3 5 4 6 9	

### Codeforces Round #586 (Div. 1 + Div. 2)

Finished

Practice



### → Virtual participation

Virtual contest is a way to take part in past contest, as close as possible to participation on time. It is supported only ICPC mode for virtual contests. If you've seen these problems, a virtual contest is not for you - solve these problems in the archive. If you just want to solve some problem from a contest, a virtual contest is not for you - solve this problem in the archive. Never use someone else's code, read the tutorials or communicate with other person during a virtual contest.

Start virtual contest

### → Practice

You are registered for practice. You can solve problems unofficially. Results can be found in the contest status and in the bottom of standings.

### → Clone Contest to Mashup

You can clone this contest to a mashup.

Clone Contest

### → Submit?

Language: GNU G++11 5.1.0

Choose file:  未选择任何文件

Be careful: there is 50 points penalty for submission which fails the pretests or resubmission (except failure on the first test, denial of judgement or similar verdicts). "Passed pretests" submission verdict doesn't guarantee that the solution is absolutely correct and it will pass system tests.

Submit

### → Problem tags

dfs and similar dp dsu graphs  
 greedy trees \*2200

No tag edit access

### → Contest materials

```
4 6
3 7
6 8
9 4
9 10
6
```

**output**

Copy

```
61
```

- Announcement #1 (en)
- Announcement #2 (ru)

---

[Codeforces](#) (c) Copyright 2010-2019 Mike Mirzayanov  
The only programming contests Web 2.0 platform  
Server time: Sep/19/2019 14:06:43<sup>UTC+8</sup> (h2).  
Desktop version, switch to [mobile version](#).  
[Privacy Policy](#)

Supported by



ITMO UNIVERSITY