

CCC '21 S4 - Daily Commute

Time limit: 3.0s Memory limit: 1G

Canadian Computing Competition: 2021 Stage 1, Senior #4

Toronto has N subway stations, numbered from 1 to N . You start at station 1, and every day, you need to reach station N to get to school.

There are W one-way walkways running amongst the stations, the i^{th} of which allows you to walk from station A_i to a different station B_i ($1 \leq A_i, B_i \leq N, A_i \neq B_i$) in 1 minute. There may be multiple walkways connecting any given pair of stations.

The subway line follows a certain route through the N stations, starting at station 1 and visiting each station once. Initially, this route consists of stations S_1, S_2, \dots, S_N , in that order. $S_1 = 1$, and S_2, \dots, S_N is a permutation of the integers $2, \dots, N$. Only one subway train runs along this route per day, departing from station 1 at 6am in the morning and taking 1 minute to reach each subsequent station. This means that, m minutes after 6am, the train will be at station S_{m+1} (or at station S_N if $m \geq N - 1$).

Over a period of D days, however, the subway line's route will keep changing. At the start of the i^{th} day, the X_i^{th} station and Y_i^{th} station ($2 \leq X_i, Y_i \leq N, X_i \neq Y_i$) in the route will be swapped. Note that, after each such change, the route will still begin at station 1 and will visit all N stations once each. Changes will carry over to subsequent days – the route will not automatically reset itself back to S_1, \dots, S_N .

On each of these D days, you'd like to determine how quickly you can get to school so you can begin learning things. On the i^{th} day, starting at 6am in the morning (after the i^{th} update to the subway line's route), you'll begin your daily trip to station N . Each minute, you may either ride the subway to its next stop (if you're currently at the same station as the train and it hasn't already completed its route), take a walkway from your current station to another one, or wait at your current station. Note that your trip begins at the same time as the train's route, meaning that you may choose to immediately ride it if you'd like to, and that you may choose to leave and then get back on the train during your trip.

Input Specification

The first line contains three space-separated integers, N , W , and D .

The next W lines each contain two space-separated integers, A_i and B_i ($1 \leq i \leq W$).

The next line contains the N space-separated integers, S_1, \dots, S_N , which form the initial permutation of stations.

The next D lines each contain two space-separated integers, X_i and Y_i ($1 \leq i \leq D$).

The following table shows how the available 15 marks are distributed.

Subtask	N	W	D
2 marks	$3 \leq N \leq 10$	$0 \leq W \leq 10$	$1 \leq D \leq 10$
2 marks	$3 \leq N \leq 200$	$0 \leq W \leq 200$	$1 \leq D \leq 200$
3 marks	$3 \leq N \leq 2\,000$	$0 \leq W \leq 2\,000$	$1 \leq D \leq 2\,000$

8 marks

 $3 \leq N \leq 200\,000$ $0 \leq W \leq 200\,000$ $1 \leq D \leq 200\,000$

Output Specification

The output is D lines, with one integer per line. The i^{th} line is the minimum number of minutes required to reach station N on the i^{th} day ($1 \leq i \leq D$).

Sample Input

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

Output for Sample Input

```
1
2
3
```

Explanation of Output for Sample Input

At the start of the first day, the subway line's route will be updated to visit stations $[1, 4, 2, 3]$, in that order. On that day, you should simply take the subway to station 4, taking 1 minute.

On the second day, the route will become $[1, 3, 2, 4]$, and you should take the subway to station 3 (taking 1 minute) and then walk to station 4 (taking 1 more minute).

On the third day, the route will become $[1, 2, 3, 4]$. One choice of optimal trip involves walking to station 2 (taking 1 minute), then boarding the train there and taking it through station 3 and finally to station 4 (taking another 2 minutes).