109 學年度
全國大專電腦軟體設計競賽
台大校內初賽

National Taiwan University

2020/09/27

Language	Version	Compile Flags	Extensions
C	gcc 9.1.0	`-O2 -std=c11 -static -lm`	`.c`
C++	g++ 9.1.0	`-O2 -std=c++17 -static -lm`	`.cc, .cpp`

Problem	Problem Name	Time Limit	Memory Limit
A	Bring Me a Polynomial	1 s	1024 MB
B	Instant Noodles	4 s	1024 MB
C	A Cool Problem	1 s	1024 MB
D	Lonely Family	4 s	1024 MB
E	Poker	1 s	1024 MB
F	LCPS	1 s	1024 MB
G	Dorm	1 s	1024 MB
H	Triple Coupon	1 s	1024 MB
I	Bookshelf	7 s	1024 MB
J	Dinner	1 s	1024 MB

This page is intentionally left blank.

A. Bring Me a Polynomial

Alice and Bob are playing a math game. The game executes as follow.

- Alice secretly chooses a polynomial $f(t) = a_{d_f}t^{d_f} + \cdots + a_1t + a_0$.
- Alice picks N numbers $x_1, \ldots, x_N$, computes $y_i = f(x_i)$, and then tells Bob the pairs (x_i, y_i) for $1 \leq i \leq n$.
- Bob tries to find a polynomial $g(t) = b_{d_g}t^{d_g} + \cdots + b_1t + b_0$ such that $g(x_i) = y_i$ for all i.

After several rounds, Bob finds that this game is boring since he can always find **infinitely many** polynomials as possible answer.

Therefore, they slightly modify the rule. Now the game goes as follow.

- Alice picks $2N$ numbers $x_1, \cdots, x_N, y_1, \cdots, y_N$. For simplicity, Alice will choose x_i, y_i as integers.
- Bob tries to find a polynomial $g(t)$ with integer coefficients, that is,

  $$g(t) = \sum_{j=0}^{d_g} b_j t^j \text{ such that } b_j \in \mathbb{Z}\ \forall\ j = 0, 1, \ldots, d_g,$$

  satisfying that $g(x_i) = y_i$ for $i = 1, 2, \ldots, N$. The degree d_g of $g(t)$ can be arbitrarily determined by Bob.

Note that there may be no solutions for Bob due to the restriction of coefficients.

The game is somewhat difficult now for Bob. Your task is to help Bob determine whether there is a solution or not.

Input

The first line of the input contains a positive integer N, indicating the number of pairs (x_i, y_i).

The second line contains N distinct integers $x_1, \ldots, x_N$.

The third line contains N integers $y_1, \ldots, y_N$.

- $1 \le N \le 5000$
- $-10^{12} \le x_i, y_i \le 10^{12}$ for $1 \le i \le N$
- $x_i \ne x_j$ for $1 \le i < j \le N$

Output

Output "Yes" (without double quotes) if there is a polynomial $g(t)$ with integer coefficients such that $g(x_i) = y_i$; output "No" (without double quotes) otherwise.

Sample Input 1

```
4
1 2 3 4
1 4 9 15
```

Sample Output 1

```
No
```

Sample Input 2

```
6
0 1 10 100 1000 10000
-256 1227 52824 3464544 -562952256 -956389520256
```

Sample Output 2

```
Yes
```

Sample Input 3

```
7
0 1 10 100 1000 10000 10001
-256 1227 52824 3464544 -562952256 -956389520256 -956389520256
```

Sample Output 3

```
No
```

Sample Input 4

```
4
10000000000 10000000001 10000000002 10000000003
-1000000000000 1000000000000 -1000000000000 800000000000
```

Sample Output 4

```
Yes
```

B. Instant Noodles

YP is fond of instant noodles. For example, he can eat instant noodles for three meals in a whole week.

As you know, instant noodles are noodles sold in a precooked and dried block with flavoring powder and seasoning oil. The flavoring is usually put in a separate packet except for cup noodles. As for cup noodles, the flavoring is often loose in the cup. Some instant noodle products are sealed, and can be reheated or eaten directly. Dried noodle blocks are designed to be cooked or soaked in boiling water before eating; however, it can also be consumed dryly.

YP is hungry now, and he wants to eat instant noodles. There is N cities (numbered from 1 to N) and M two-way roads in the country YP lives in. The i-th road $(1 \leq i \leq M)$ connects city u_i and v_i. YP will consume l_i calories after going through the i-th road once. It is guaranteed that each city is reachable from any other city. YP lives in the city 1. In order to eat instant noodles, YP needs to buy some instant noodles first, then get boiling water from some water dispenser. There are only S shops selling instant noodles in the country. Those shops are located in cities $a_1, a_2, \ldots, a_S$. Also, there are only W water dispensers in the country. The water dispensers are located in cities $b_1, b_2, \ldots, b_W$.

Now, please help YP to calculate that, in order to eat his instant noodles, what is the minimal calories that YP must consume. Note that YP must get boiled water **after** he buys instant noodles. Walking in the same city doesn't consume calories. For example, if there are instant noodles shop and water dispenser in city 1, then YP doesn't need to consume any calories!

Input

The first line of the input contains four integers N, M, S, W, which indicate the number of cities in the country, the number of roads in the country, the number of instant noodles shops in the country, and the number of water dispensers in the country, respectively.

The second line contains S integers $a_1, a_2, \ldots, a_S$, which indicate the locations of the instant noodles shops.

The third line contains W integers $b_1, b_2, \ldots, b_W$, indicates the the locations of water dispensers.

In the next M lines, each line contains three integers u_i, v_i, and l_i denoting there is a road which connects city u_i and v_i, and YP will consume l_i calories after going through the road.

- $1 \leq N \leq 5 \times 10^5$
- $N - 1 \leq M \leq 5 \times 10^5$
- $1 \leq S, W \leq N$
- $1 \leq a_i \leq N$, all a_i are distinct.
- $1 \leq b_i \leq N$, all b_i are distinct.
- $1 \leq u_i, v_i \leq N, u_i \neq v_i$
- There may be roads connecting the same pair of cities
- $1 \leq l_i \leq 10^9$
- It is guaranteed that each city is reachable from any other city

Output

Output the minimal calories that YP needs to consume in one line.

Sample Input 1	Sample Output 1
6 6 2 2 2 4 3 5 1 6 5 1 2 3 2 3 7 1 5 1 5 4 2 1 6 4	5

Sample Input 2	Sample Output 2
1 0 1 1 1 1	0

C. A Cool Problem

Given a string $S_1S_2\dots S_N$ with length N.

A pair (l, r) is good if the following conditions are satisfied:

- $1 \leq l \leq r \leq N$
- If one reverse the substring $S_lS_{l+1}\dots S_r$ in S, the resulting string is a palindrome.

Please count the number of good pairs.

Input

The first line of the input contains one integer N indicating the length of S.

The second line contains a string S.

- $1 \leq N \leq 10^5$
- $|S| = N$
- S only contains lowercase letters.

Output

Output the number of good pairs in one line.

Sample Input 1	Sample Output 1
7 ababbaa	2

Sample Input 2	Sample Output 2
13 ababbababcbaa	1

This page is intentionally left blank.

D. Lonely Family

There is a beautiful kingdom. The kingdom contains N towns and $N-1$ roads connecting the towns. The towns are labelled $1, 2, \ldots, N$. You can travel between any two towns along the roads.

There are Q families living in this kingdom. For each family, every member lives in different towns. The lonely value of a member in a family is the sum of the shortest distance from the town where this member lives, to the towns where the other family members lives. The lonely value of a family is the sum of the lonely values of all members.

The i-th family has $r_i - l_i + 1$ members. The j-th member in the i-th family lives in the town $l_i + j - 1$. So, the members in the i-th family live in the towns $l_i, l_i + 1, \ldots r_i$.

The king of the kingdom is worried about that there are some families being too lonely. So the king asks you to calculate the lonely value for each family.

Input

The first line of the input contains an integer N, indicating the number of towns in the kingdom.

The following $N-1$ lines describe the roads in the kingdom. The i-th line of the following $N-1$ lines contains three space-separated numbers u_i, v_i, w_i, indicating there is a road connecting town u_i and town v_i with length w_i.

Then, the following line contains an integer Q.

The following Q lines describe the families in the kingdom. The i-th line contains two space-separated integers l_i, r_i, describing the i-th family.

- $1 \leq N \leq 20000$
- $1 \leq w_i \leq 10^4$
- $1 \leq u_i, v_i \leq N, u_i \neq v_i$
- $1 \leq Q \leq 20000$
- $1 \leq l_i \leq r_i \leq N$

Output

Output Q lines, the i-th line contains the lonely value of the i-th family.

Sample Input 1	Sample Output 1
5 1 2 3 2 3 4 2 4 5 4 5 7 5 1 3 2 4 3 5 1 4 1 5	28 36 64 72 172

Sample Input 2	Sample Output 2
1 1 1 1	0

E. Poker

YP loves to play poker games, and he often plays such games with Yen-Jen in their free time.

Poker is a collection of card games in which players wager over which hand is the best according to some specific hand ranking rules. Furthermore, Poker games vary in deck configurations, the number of cards in play, the number of cards dealt face up or face down, and the number or cards shared by all players. Hence, all of them involve one or more rounds of betting.

Now, YP has N cards on a table. The number of the cards from the top of the deck to the bottom of the deck is $a_1, a_2, \ldots, a_N$ respectively, where $a_1, a_2, \ldots, a_N$ is a permutation of $\{1, 2, \ldots, N\}$. Specifically, N is an even number.

YP discovered an interesting gameplay today and played with Yen-Jen. The rule of the game goes as follows.

- Before each round, YP can decide either to play 2 cards or 4 cards. If YP selects to play 4 cards, then there must be at least four cards left in the deck.
- If YP decides to play 2 cards, the round goes as follows:
  1. YP gets one card from the top of the deck.
  2. Yen-Jen gets one card from the top of the deck.
  3. If the number on YP's card is greater than the number on Yen-Jen's card, then YP wins.
  4. Otherwise, YP loses.
- If YP decides to play 4 cards, the round goes as follows:
  1. YP gets one card from the top of the deck. Let x be the number of this card.
  2. Yen-Jen gets one card from the top of the deck. Let y be the number of this card.
  3. YP gets one card from the top of the deck. Let z be the number of this card.
  4. Yen-Jen gets one card from the top of the deck. Let w be the number of this card.
  5. If $x > y$ and $z > w$, then YP wins.
  6. If $x < y$ and $z < w$, then YP loses.
  7. Otherwise, this round ties.

YP knows one magician friend: Meow. Meow can cast at most K magics. In each magic, Meow will ask YP to select two integers x, y, and Meow will swap a_x, a_y secretly.

YP wants to maximize **"the number of rounds he wins"$-$"the number of rounds he loses"**. Can you help him to find out this value?

Also, it is guaranteed that for any two different odd indices i and j, $min(a_i, a_{i+1}) < min(a_j, a_{j+1}) < max(a_j, a_{j+1}) < max(a_i, a_{i+1})$ do not hold.

Input

The first line of the input contains one integer T, denoting the number of test cases.

For each test case, the first line of the test case contains two integers N and K, denoting the number of cards in the deck and the number of times that Meow can cast magic.

The second line of each test case contains N integers $a_1, a_2, \ldots, a_N$, denoting the cards in the deck from top to bottom.

- $1 \le T \le 5 \times 10^5$
- $1 \le N \le 5 \times 10^5$
- $0 \le K \le 5 \times 10^5$
- $1 \le a_i \le N$
- $a_1, a_2, \ldots, a_N$ is a permutation of $\{1, 2, \ldots, N\}$
- The summation of N in the T testcases will not exceed 5×10^5
- It is guaranteed that for any two different odd indices i and j, $min(a_i, a_{i+1}) < min(a_j, a_{j+1}) < max(a_j, a_{j+1}) < max(a_i, a_{i+1})$ do not hold.

Output

For each test case, output the maximum possible value of **"the number of rounds YP wins"**$-$**"the number of rounds YP loses"** in one line.

Sample Input 1	Sample Output 1
3 6 6 1 2 3 4 5 6 2 0 1 2 4 4 4 3 2 1	3 -1 2

F. LCPS

Brief description: Given two random-generated sequences a and b of length N and M respectively, please calculate the length of the LCPS of these two sequences.

Today, TDD, Utaha, and Yen-Jen attended a CV (Computer Vision) course.

It's an **excellent** course and can help you to promote your creativity and potential.

For example, just after the class, Yen-Jen found a cool problem – **LCPS**, which stands for the longest common palindromic subsequence.

In this problem, you have to find out **the longest common subsequence (LCS) of two random-generated sequences which is a palindrome**.

The following are definitions of LCS and palindrome (if you have already known, you can skip):

The **longest common subsequence (LCS)** problem is the problem of finding the longest subsequence common to all given sequences (in most time, there are just two sequences). It differs from the longest common substring pro blem: unlike substrings, a subsequence is not required to occupy consecutive positions within the original sequences. The longest common subsequence problem is a classic computer science problem, is the basis of data comparison programs such as the diff utility, and has applications in computational linguistics and bioinformatics. It is also widely used by revision control systems such as Git for reconciling multiple changes made to a revision-controlled collection of files.

For instance, considering the sequences `"ABCD"` and `"ACBAD"`. There are 5 common subsequences with length 2: `"AB"`, `"AC"`, `"AD"`, `"BD"`, and `"CD"`; 2 common subsequences with length 3: `"ABD"` and `"ACD"`; and no longer common subsequences. So `"ABD"` and `"ACD"` are their longest common subsequences.

A **palindrome** is a word, number, phrase, or other sequence of characters which reads the same backward as forward, such as madam, TDDxDDT.

Input

The first line of the input contains two integers N, M indicating the length of sequences a, b respectively.

The second line contains N space-separated integers $a_1, a_2, \ldots, a_N$.

The third line contains M space-separated integers $b_1, b_2, \ldots, b_M$.

- $1 \leq N, M \leq 400$
- $1 \leq a_i, b_i \leq 100$
- $a_1, \ldots, a_N, b_1, \ldots, b_M$ are generated independently and uniformly from $[1, 2, \ldots, 100]$

Output

Print one integer – the length of the LCPS of a and b.

Sample Input 1	Sample Output 1
6 5 4 98 23 56 54 4 32 4 98 12 4	3

G. Dorm

Welcome to Cute Cat Dorm.

There are many cats living in this dormitory.

Cute Cat Dorm can be described as an $N \times M$ grid, and each grid can accommodate a cat.

However, there are exactly K broken grids. No cat can live in a broken grid.

Furthermore, in order to make the dorm cute enough, there are strict requirements on the total number of the cats that should live in each row and column. To be more specific, the restriction of i-th row is r_i, which means the total number of the cats that live in the i-th row should be **at least** r_i. Similarly, the restriction of i-th column is c_i, which means the total number of the cats that live in the i-th column should be **at least** c_i.

Now, your task is to calculate the **minimum** possible number of the cats that live in this dormitory, such that the $N + M$ requirements are all satisfied.

Input

The first line contains three integers N, M, and K, indicating the size of the dorm and the number of broken grids. In the i-th line of the following K lines, there are two integers x_i and y_i, indicating that the grid (x_i, y_i) is broken. The second last line contains N integers $r_1, r_2, \ldots, r_N$, indicating the restrictions of rows. The last line contains M integers $c_1, c_2, \ldots, c_M$, indicating the restrictions of columns.

- $1 \leq N, M \leq 100$
- $0 \leq K \leq N \times M$
- $1 \leq x_i \leq N$
- $1 \leq y_i \leq M$
- $0 \leq r_i \leq M, \ i = 1, \ldots, N$
- $0 \leq c_i \leq N, \ i = 1, \ldots, M$

Output

If there exists at least one possible configuration satifying all the restrictions, please print one integer – the minimum number of the cats that live in this dormitory.

Otherwise, please print `"Meow!"` (without double quotes).

Sample Input 1	Sample Output 1
3 3 0 1 1 1 1 1 1	3

Sample Input 2	Sample Output 2
2 3 5 1 1 1 2 1 3 2 1 2 2 1 0 0 0 0	Meow!

Sample Input 3	Sample Output 3
3 2 2 1 1 3 1 0 2 0 0 3	4

H. Triple Coupon

In this year, everything is tripled!

If you earn 1 coin, you will get 2 free coins as bonus. If you buy one bento, you will get two more to make sure you won't get hungry.

Currently, you have spent all of your money and can't buy any bento. To prevent you from starving, Eddy offers you a chance to earn money in order to buy bentos. You will be given **4** sticks and need to create a triangle from them. You don't need to use all of the sticks. After you successfully create the triangle, Eddy will give you as much money as the **triple** of the area of the formed triangle.

What's the most possible money you can get from Eddy?

Input

The first line contains one integer T indicating the number of the scenarios.

Each of the following T line contains **4** space-separated integers A, B, C, D indicating the length of the sticks.

- $1 \leq T \leq 10^4$
- $1 \leq A, B, C, D \leq 10^4$

Output

For each scenario, output one line with one number indicating the most possible money you can earn.

The answer will be judged as correct if the relative or absolute error is less than 10^{-6}.

Sample Input 1	Sample Output 1
3 3 4 5 1021 1 2 4 5 1 1 1 1	18.00000000 18.00000000 1.29903811

I. Bookshelf

Yen-Jen has a magical bookshelf. The bookshelf has a single layer, and there are N books in a row on this shelf. Each book has its own **height** h_i and **type** t_i ($t_i \in \{1, 2, 3\}$).

Because the bookshelf is too messy, Yen-Jen wants to rearrange the books on the shelf. Furthermore, he decides to spend Q days to rearrange the books due to the large number of books.

In i-th day, Yen-Jen will rearrange the books from L_i-th book to R_i-th book on the shelf. It is guaranteed that $R_i - L_i + 1$ is an even number.

However, there are some constraints for each book. If a book is of type **1**, then it must be placed in the left half of the range. That is to say, it must be placed in the interval $\left[L_i, \frac{L_i+R_i-1}{2}\right]$. If a book is of type **2**, then it must be placed in the right half of the range. To be more specific, it must be placed in the interval $\left[\frac{L_i+R_i+1}{2}, R_i\right]$. If a book is of type **3**, then it can be placed in anywhere in the range. In other words, it can be placed in anywhere in $[L_i, R_i]$.

After rearranging, Yen-Jen want to minimize $v_i = \sum_{j=L_i}^{R_i-1} |h_j - h_{j+1}|$. Can you tell Yen-Jen what is the minimum possible value of v_i?

Last but not least, you may wonder why the bookshelf is a magical shelf, right? It is because that after Yen-Jen rearranges the books today, they will restore to their original position tomorrow.

Input

The first line of the input contains an integer T denoting the number of test cases.

For each test case, the first line of the input contains two space-separated integers N and Q, denoting the number of books on the bookshelf and the number of days that Yen-Jen decides to rearrange.

The second line of the test case contains N integers $h_1, h_2, \ldots, h_N$, denoting the height of each book.

The third line of the test case contains N integers $t_1, t_2, \ldots, t_N$, denoting the type of each book.

In the next Q lines of the test case, the i-th line contains two integers L_i, R_i denoting the range Yen-Jen wants to rearrange.

- $1 \le T \le 3 \times 10^5$
- $2 \le N \le 3 \times 10^5$
- $1 \le Q \le 3 \times 10^5$
- $1 \le h_i \le 10^9$
- $1 \le t_i \le 3$
- $1 \le L_i < R_i \le N$
- $R_i - L_i + 1$ is guaranteed to be even number
- The summation of N in the T testcases will not exceed 3×10^5
- The summation of Q in the T testcases will not exceed 3×10^5

Output

For each test case, please output Q lines. In i-th line of the test case, if there exists a way to rearrange the books for i-th day, please output the minimum possible value of v_i. Otherwise, please output -1.

Sample Input 1

```
2
5 3
3 6 5 4 3
2 2 2 1 3
1 4
2 5
3 4
4 1
1 4 5 3
1 3 3 2
1 4
```

Sample Output 1

```
-1
3
1
6
```

J. Dinner

You are the chef of a restaurant. All your customers want to enjoy their meals as soon as possible. However, as a disciplined chef, you insist that you only make one dish at a time.

For customer i, you will spend exactly t_i minutes for his or her meal. The only thing you can do to improve your customers' dining experience is to optimize your cooking order. To be more specific, for customer i, there is a weight u_i indicating his or her importance. Moreover, your customers come in M groups, and there is also a weight v_j for group j.

You want to optimize the weighted sum of the waiting time for your customers. That is to say, you want to minimize $\sum_i u_i r_i + \sum_j v_j s_j$, where r_i is the waiting time for customer i and s_j is the waiting time for group j. The waiting time of a group is defined as the longest waiting time of the person in that group.

Input

The first line contains an integer M, indicating the number of groups.

For each group, the first line contains two integers v_j and N_j, indicating the weight and the number of members of this group. Each of the following N_j lines contains two integers u_i and t_i, indicating the weight for i-th customer in the group and the preparation time for his or her meal.

- $1 \leq M \leq 1000$
- $1 \leq v_j, N_j \leq 1000$
- $1 \leq u_i, t_i \leq 100$

Output

Output a single line with the minimum possible weighted sum of the waiting time of customers.

Sample Input 1

```
1
5 3
1 1
1 1
1 1
```

Sample Output 1

```
21
```

Sample Input 2

```
2
4 2
1 1
1 1
1 2
5 1
1 1
```

Sample Output 2

```
30
```