Activities Google Chrome Wed 14:20

IITG Authentication Keep | InterviewBit

interviewbit.com/test/603a162065/?slug=c036018f1467d734784c9f9e570f503215de17f2#/problem_1

Incognito

IITG MTP Welcome to CCD Algorithms - G... DP LEET GitHub - trier-...

InterviewBit | Media.net - IIT Guwahati Online Coding Test 2019-20 | 00 Hr : 39 min : 37 sec | 1 / 3 Attempted | End Test | Issues ?

S1

Q 1

Q 2

Q 3

Biggest square

Flag Question

You are given a **matrix A** of **R** rows and **C** columns. Each cell of the matrix is colored either black or white.
A black cell is denoted by **1** and a white cell is denoted by **0**.

You are given **Q** queries. In each query, you will be given a **coordinate of a cell (X, Y)** and an integer **K**.
Here **X** denotes the row number and **Y** denotes the column number.

For each query, you are required to tell the maximum possible area of a sqaure that has **(X, Y)** as a center and contains at most **K** black cells.

Note:

```
1. For a square to have center at (x, y), it must have odd length.
2. Rows are numbered from top to bottom and columns are numbered from left to right.
```

Input Format

```
The first argument is a matrix A.
The second argument is an integer array B denoting the values of X for the coordinates of each query.
The third argument is an integer array C denoting the values of Y for the coordinates of each query.
The fourth argument is an integer array D denoting the values of K for each query.
```

Output Format

```
Return an array of size Q having answers for each query.
```

Constraints

```
1 <= R, C <= 3000
1 <= X <= R
1 <= Y <= C
1 <= K <= R * C
1 <= Q <= 5 * 10^5
A[i][j] = 0 or 1
```

For Example

```
Input 1:
    R = 4, C = 7

    A = 0 1 0 1 1 0 1
        0 1 0 1 0 1 0
        0 1 0 1 0 1 1
        0 1 1 1 1 1 0

    Q = 1
        X = 3, Y = 6, K = 5
Output 1:
    9
```

Maximum Subtree

Flag Question

You are given a tree of **A** nodes having **A-1** edges. Each node is numbered from **1** to **A** where **1** is the root of the tree.

You are given **Q** queries. In each query, you will be given a **unique** integer **j**. You are required to remove the **j'th** numbered edge from the tree.

This operation will divide a tree into two different trees.

For each query, once you perform the remove operation, you are asked to tell the maximum size among all the sizes of the trees present till that query.

Note:

1. Once an edge is removed, it will be considered removed for all the further queries.
2. It is guaranteed that each edge will be pointing to exactly two different nodes of the tree.
3. Edges are numbered from **1** to **A-1**. Please read the input format for more clarification.

Input Format

```
The first argument is an integer A denoting the number of nodes.
The second and third arguments are the integer arrays B and C where for each i (0 <= i < A-1), i denotes the (i+1)th edge and B[i] and C[i] are the nodes connected by it.
The fourth argument is an integer array D of distinct elements where D[i] denotes the number of the edge to be removed for the ith query.
```

Output Format

```
Return an array of answers for each query.
```

Constraints

```
2 <= A <= 10^5
1 <= B[i], C[i] <= A
1 <= D[i], Q <= A-1
```

Example

```
Input 1:
    A = 5
    B = 1 3 3 5
    C = 3 2 4 1
    D = 1 3

Output 1:
    3 2

Explanation:
    Initial tree:
        1
       / \
```

```
1 <= D[i], Q <= A-1
```

Example

```
Input 1:
    A = 5
    B = 1 3 3 5
    C = 3 2 4 1
    D = 1 3

Output 1:
    3 2

Explanation:
    Initial tree:
        1
       / \
      3   5
     /\
    2  4

    Query 1:
    Remove edge number 1: 1-3
        1
         \
      3   5
     /\
    2  4

    Obtained Trees:-
        Node 2, Node 3, Node 4 => Size 3
        Node 1, Node 5 => Size 2

    Max Size = 3

    Query 2:
    Remove edge number 3: 3-4
        1
         \
      3   5
     /
    2  4
    Obtained Trees:-
        Node 2, Node 3 => Size = 2
        Node 4 => Size = 1
        Node 1, Node 5 => Size = 2

    Max Size = 2
```

Minimum cost to visit the museum

Flag Question

Shadowland is a beautiful city with **N** museums which are connected with bidirectional roads. The city can be represented as a graph with **N** nodes representing museums and **M** edges representing roads.

There is a cost of traveling each road in order to move from the front of one museum to that of the other. However, there is an additional cost one has to pay if and only if he decides to enter the museum. The cost of entering the museums need not be the same.

For each museum **i**, Abhishek wants to know the minimum total amount he has to pay if he starts his journey from the front of the **ith** museum and enters any one museum.

Note:

```
1. Abhishek may or may not enter the same museum he is starting from.
2. The graph may or may not be fully connected.
3. There may be multiple roads between the same pair of museums as well.
4. There may be roads which connect the museum to itself.
5. While traveling the roads he will not enter any museum that he encounters on his way till he finally enters the museum such that the total cost of his journey is minimized.
```

Given an array of integers **A** of size **N** for which **A[i]** represents the cost of entering the **i'th** museum. And three more integer arrays **B**, **C** and **D**. **B[i]**, **C[i]** and **D[i]** represents the **i'th** road i.e. there is an edge from **B[i]** to **C[i]** with edge weight **D[i]**.

Find and return the integer array in which the **i'th** element represents the minimum cost Abhishek has to pay in order to visit any museum if he is standing in front of the **i'th** museum.

Input Format

```
The first argument given is the integer array A.
The second argument given is the integer array B.
The third argument given is the integer array C.
The fourth argument given is the integer array D.
```

Output Format

```
Return an integer array in which the i'th element represents the minimum cost Abhishek has to pay in order to enter any museum if he starts from the front of the i'th museum.
```

Constraints

```
1 <= N <= 100000
1 <= M <= 200000
1 <= A[i], D[i] <= 10^9
1 <= B[i], C[i] <= N
```

For Example

```
1. Abhishek may or may not enter the same museum he is starting from.
2. The graph may or may not be fully connected.
3. There may be multiple roads between the same pair of museums as well.
4. There may be roads which connect the museum to itself.
5. While traveling the roads he will not enter any museum that he encounters on his way till he finally enters the museum such that the total cost of his journey is minimized.
```

Given an array of integers **A** of size **N** for which **A[i]** represents the cost of entering the **i'th** museum. And three more integer arrays **B**, **C** and **D**. **B[i]**, **C[i]** and **D[i]** represents the **i'th** road i.e. there is an edge from **B[i]** to **C[i]** with edge weight **D[i]**.

Find and return the integer array in which the **i'th** element represents the minimum cost Abhishek has to pay in order to visit any museum if he is standing in front of the **i'th** museum.

Input Format

```
The first argument given is the integer array A.
The second argument given is the integer array B.
The third argument given is the integer array C.
The fourth argument given is the integer array D.
```

Output Format

```
Return an integer array in which the i'th element represents the minimum cost Abhishek has to pay in order to enter any museum if he starts from the front of the i'th museum.
```

Constraints

```
1 <= N <= 100000
1 <= M <= 200000
1 <= A[i], D[i] <= 10^9
1 <= B[i], C[i] <= N
```

For Example

```
Input 1:
    A = [1, 2, 3, 1, 5]
    B = [1, 2, 3, 4]
    C = [2, 3, 4, 5]
    D = [1, 1, 1, 1]
Output 1:
    [1, 2, 2, 1, 2]

Input 2:
    A = [5, 17, 100, 11]
    B = [2, 3]
    C = [1, 4]
    D = [5, 50]
Output 2:
    [5, 10, 61, 11]
```