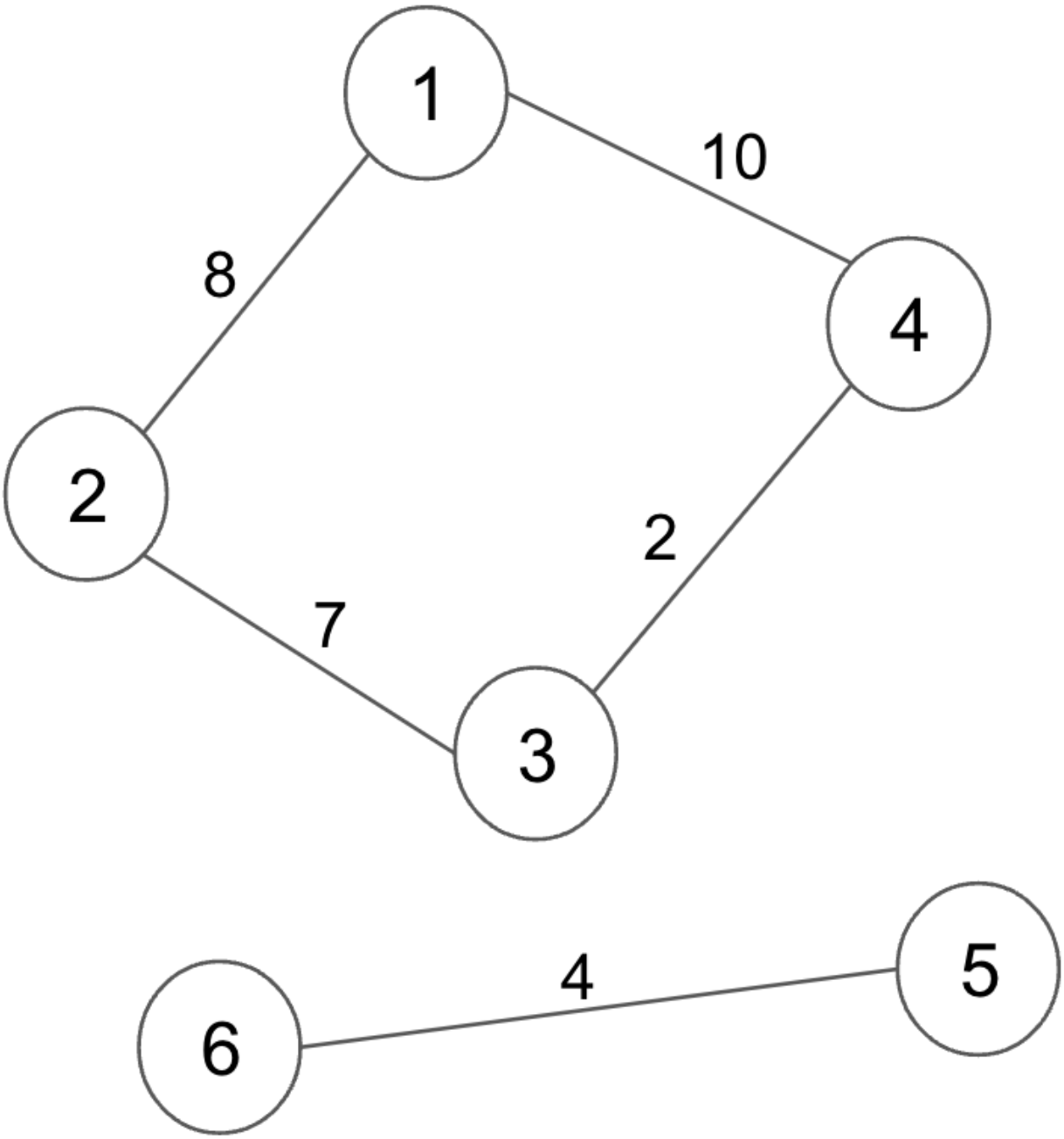
Problem D - Shortest Path Game

Time Limit	Memory Limit
1 second (See Below)	512 MB

Description

Albert enjoys the game of finding shortest paths in an undirected graph. Yet because it's boring to find shortest paths in a given graph, he has come up with a new game where he would be adding a new edge or removing a previously added edge so as to keep changing the graph while computing shortest paths.

First, he would create a graph G with N vertices and M edges where the vertices are numbered as 1,2,...,N. The i-th edge denoted as (Xi,Yi,Ci) means that it connects two vertices Xi and Yi and its weight is Ci. For instance, the image below shows a graph with N=6, M=5, X=[1,2,1,4,5], Y=[2,3,4,3,6], and C=[8,7,10,2,4]. Notice that edges are undirected. There can exist multiple edges between the same pair of vertices (see sample test cases), but every edge will satisfy Xi≠Yi.



After creating the initial graph G, Albert would perform Q operations on the graph where there exist 3 types of operations. Let $R_k \in \{1,2,3\}$ represent the type of the k-th operation.

- Operation Type 1: Compute the score of the current graph. The score of a graph is defined as the sum of the shortest path between

each pair of vertices (i,j) with i≠j; if there is no path between (i,j), then such pair is excluded (ignored) in calculating the score of the graph. Albert would write down the score of a graph on paper.

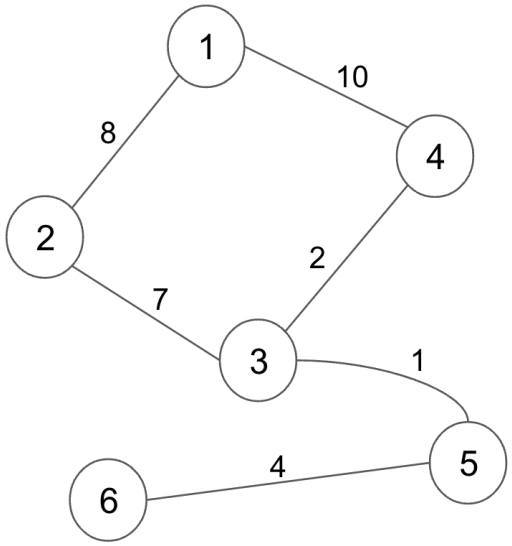
- Operation Type 2: Add to the current graph an edge with weight Wk that connects two vertices Ak and Bk. Even if there already exists an edge between the vertices, add a new edge.
- Operation Type 3: Remove from the current graph the latest edge that was added through Operation Type 2 above. Yet, if there is no edge in the current graph that was added through Operation Type 2, then this operation has no effect.

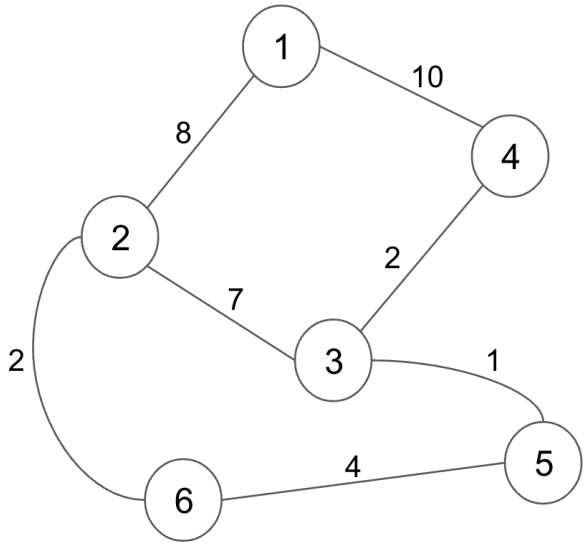
Because Ak, Bk, Wk are defined only when Rk=2, let us define Ak=Bk=Wk=0 when Rk≠2 for convenience.

For instance, suppose Albert will perform Q=9 operations to the initial graph above, and the operation types are given by R=[1,2,1,2,1,3,1,3,1], and that A=[0,3,0,2,0,0,0,0,0], B=[0,5,0,6,0,0,0,0,0], and W=[0,1,0,2,0,0,0,0,0].

- R1=1: To calculate the score of the graph, Albert must compute the shortest path distance for the following pairs of vertices: (1,2),(1,3),(1,4),(2,3),(2,4),(3,4),(5,6). All other pairs have no paths between them -- for instance, there is no path between (1,5) or between (4,5). The shortest path distance for the seven pairs are, in the same order, 8,12,10,7,9,2,4, and therefore the score of the graph is 52.
- R2=2: Albert will add an edge with A2=3, B2=5, which connects between (3,5) whose weight is W2=1. The image below on the left shows the new graph after adding the edge.

- R3=1 : Similar to the first operation, the score of the graph must be computed. There now exist paths between one of 1, 2, 3, 4 and one of 5, 6, and thus such paths must be included. The score is 118.
- R4=2 : Albert will now add an edge with weight 2 that connects (2,6) , and the image below on the right shows the new graph.
- R5=1 : By using the newly added edges, some pairs of vertices now have shorter paths between them. For instance, the shortest path distance between (2,6) was 12 before, but it is now 2. The score of this graph is 99.
- R6=3 : Albert will delete the last edge that was added which connects (2,6) from the graph, resulting in a new graph depicted in the below image on the left.
- R7=1 : The score of this graph is 118 as before.
- R8=3 : Albert will delete the last edge that was added which connects (3,5) from the current graph. Note that the edge connecting (2,6) has already been removed at this point. The resulting graph is the same as the initial graph with five edges.
- R9=1 : The score of this graph is 52 as before.
- After performing all operations, Albert will have written the scores of the graphs on a piece of paper as "`52 118 99 118 52`".





Given N,M,Q,X,Y,C as well as R,A,B output the scores of the graph (due to Operation Type 1) -- Albert needs your help to ensure that his answers are correct.

Input

The first line of the input will contain T, the number of test cases.

The first line of each test case will contain N,M and Q, separated by whitespace.
The i-th of the next M lines will contain 3 integers Xi, Yi, Ci separated by whitespace.
The k-th of the next Q lines will contain the type of the k-th operation -- and if its type is 2, then additional parameters to describe the operation. The first integer in each line will be Rk that represents the type. If and only if Rk=2, the same line will contain 3 additional integers, Ak, Bk and Wk.

Output

Output the scores of the graphs for each test case, separated by whitespace, in each line.

Limit

- $1 \le T \le 10$
- $2 \le N \le 200$
- $1 \le M \le 50000$
- $3 \le Q \le 200$
- For each i with $1 \le i \le M$:
  - $1 \le Xi, Yi \le N$
  - $Xi \neq Yi$
  - $1 \le Ci \le 10^6$
- For each k with $1 \le k \le Q$, $Rk \in \{1,2,3\}$.
- For each test case, there is at least one k with $Rk=1$, one k with $Rk=2$, and one k with $Rk=3$.
- For each k with $Rk=2$:
  - $1 \le Ak, Bk \le N$
  - $Ak \neq Bk$
  - $1 \le Wk \le 10^6$

Sample Input 1 Copy

```
4
6 5 9
1 2 8
2 3 7
1 4 10
4 3 2
```

```
5 6 4
1
2 3 5 1
1
2 2 6 2
1
3
1
3
1
4 1 16
1 2 10
1
2 1 2 100
1
2 3 4 10
1
2 2 3 20
1
2 2 3 5
2 1 2 5
1
3
1
3
1
```

```
3
1
4 1 5
1 2 10
2 1 2 100
3
1
3
1
4 1 11
1 2 10
1
2 1 2 5
1
2 2 1 1
1
3
1
3
1
3
1
```

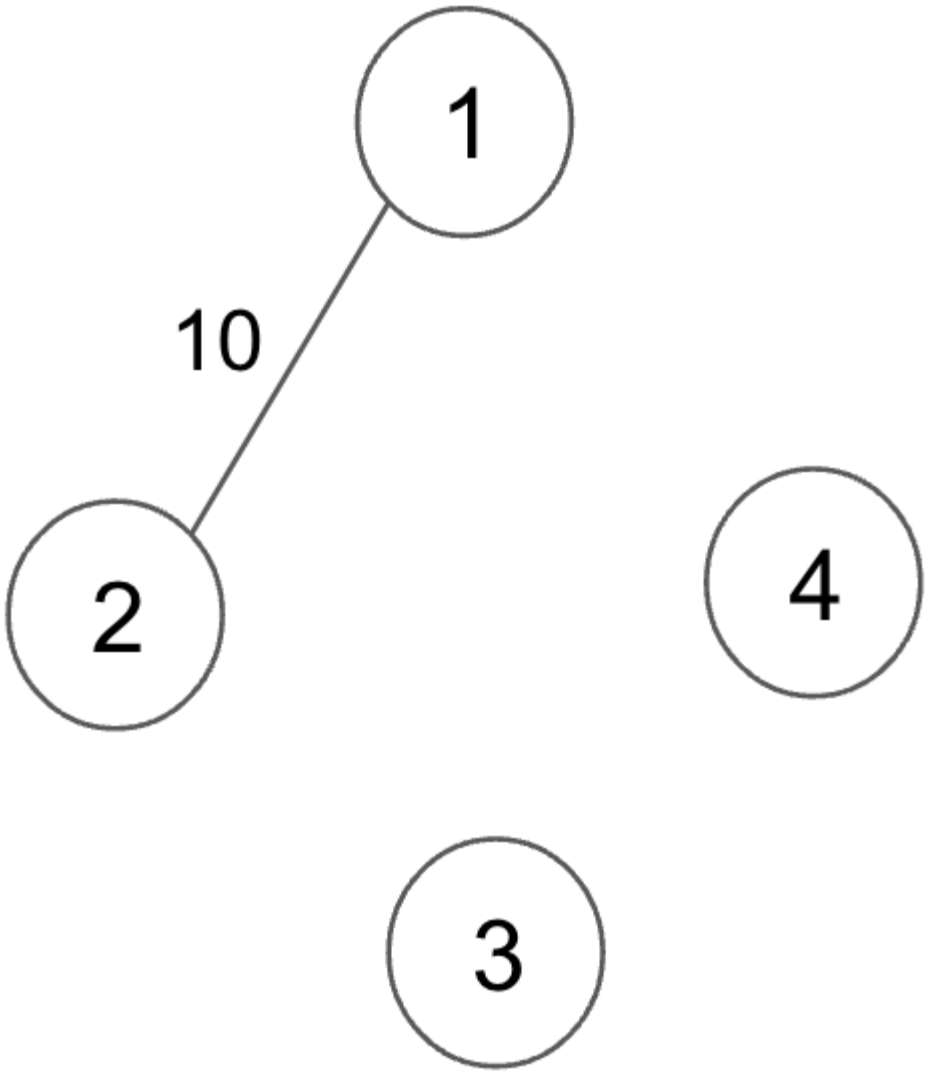
Sample Output 1 Copy

```
52 118 99 118 52
10 10 20 140 65 80 140 20
```

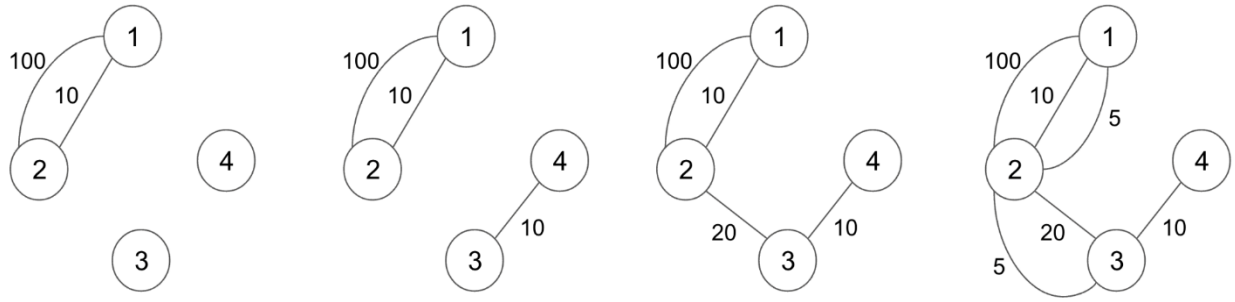
```
10 10
10 5 1 5 10 10
```

Case 1: Discussed in the problem statement.

Case 2: The intial graph looks like the following with 1 edge.



Afterwards, each time an edge is added, the graph will change as shown below. Note that there may exist multiple edges between a pair of vertices.



Case 3: The initial graph is the same as Case 2's initial graph.

- R1=2 : Add an edge with weight 100 that connects (1,2). The new graph is shown in the left image above.
- R2=3 : Remove the edge with weight 100 that connects (1,2) -- this leads to the initial graph.
- R3=1 : The score of the graph is 10.
- R4=3 : This operation has no effect because there is no edge to be removed. The only edge in the current graph was given as part of the initial graph (and not an edge added by Operation Type 2), and thus it cannot be removed.
- R5=14 : The score of the graph is still 10.

Case 4: No explanation provided.

Time Limit

- Java 8: 2 seconds
- PyPy3: 5 seconds