

# Path Choosing

Input file:            **standard input**  
Output file:           **standard output**  
Time limit:            2 seconds  
Memory limit:         256 megabytes

You're given a directed graph. Each edge of the graph has its *weight* and *priority*. Let us say that one path is *lexicographically less* than the other path if the sequence of edge priorities for the first path is lexicographically less than the similar sequence for the second path.

You have to write a program that processes queries of two types:

1. Find total weight of edges in the path that is  $k$ -th lexicographically among all paths starting at vertex  $v$  (empty path is included in this count), or determine that such path does not exist.
2. Change the weight of the edge  $e$  to  $w$ .

## Input

The first line of input contains three integers  $n$ ,  $m$  and  $q$  — the number of vertices, edges and queries respectively ( $1 \leq n, q \leq 10^5$ ;  $1 \leq m \leq 10^6$ ).

Each of the next  $m$  lines contains four integers describing respective edge:  $x_i$ ,  $y_i$ ,  $p_i$  and  $c_i$  — start vertex index, target vertex index, priority and weight ( $1 \leq x_i, y_i \leq n$ ;  $1 \leq p_i \leq m$ ;  $0 \leq c_i \leq 10^6$ ). All priorities are distinct.

Note that loops and multiple edges are allowed.

Each of the next  $q$  lines describes a query. The first number  $t_i$  describes the query type.

If  $t_i = 1$  then next follow two integers  $v_i$  and  $k_i$  — the vertex index and the path index ( $1 \leq v_i \leq n$ ;  $1 \leq k_i \leq 10^{12}$ ).

If  $t_i = 2$  then next follow two integers  $e_i$  and  $w_i$  — the edge index and new weight ( $1 \leq e_i \leq m$ ;  $0 \leq w_i \leq 10^6$ ).

## Output

For each query of type 2 output a single number — answer to the query in a separate line.

## Examples

standard input	standard output
3 4 6	0
1 2 2 1	1000
2 2 4 10	121
2 3 3 100	101
1 3 1 1000	-1
1 1 1	
1 1 2	
1 1 8	
2 2 0	
1 1 8	
1 3 2	