C. King's Path

time limit per test 2 seconds
memory limit per test 256 megabytes
input standard input
output standard output

The black king is standing on a chess field consisting of 10^9 rows and 10^9 columns. We will consider the rows of the field numbered with integers from 1 to 10^9 from top to bottom. The columns are similarly numbered with integers from 1 to 10^9 from left to right. We will denote a cell of the field that is located in the i-th row and j-th column as (i, j).

You know that some squares of the given chess field are *allowed*. All allowed cells of the chess field are given as n segments. Each segment is described by three integers $r_i, a_i, b_i (a_i \le b_i)$, denoting that cells in columns from number a_i to number b_i inclusive in the r_i-th row are allowed.

Your task is to find the minimum number of moves the king needs to get from square $(x_0, y_0)$to square (x_1, y_1), provided that he only moves along the allowed cells. In other words, the king can be located only on allowed cells on his way.

Let us remind you that a chess king can move to any of the neighboring cells in one move. Two cells of a chess field are considered neighboring if they share at least one point.

Input

The first line contains four space-separated integers x_0, y_0, x_1, y_1 $(1 \le x_0, y_0, x_1, y_1 \le 10^9)$, denoting the initial and the final positions of the king.

The second line contains a single integer n $(1 \le n \le 10^5)$, denoting the number of segments of allowed cells. Next n lines contain the descriptions of these segments. The i-th line contains three space-separated integers r_i, a_i, b_i $(1 \le r_i, a_i, b_i \le 10^9, a_i \le b_i)$, denoting that cells in columns from number a_i to number b_i inclusive in the r_i-th row are allowed. Note that the segments of the allowed cells can intersect and embed arbitrarily.

It is guaranteed that the king's initial and final position are allowed cells. It is guaranteed that the king's initial and the final positions do not coincide. It is guaranteed that the total length of all given segments doesn't exceed 10^5.

Output

If there is no path between the initial and final position along allowed cells, print -1.

Otherwise print a single integer — the minimum number of moves the king needs to get from the initial position to the final one.

Examples

input Copy

```
5 7 6 11
3
5 3 8
6 7 11
5 2 5
```

output Copy

```
4
```

input Copy

```
3 4 3 10
3
3 1 4
4 5 9
3 10 10
```

output Copy

```
6
```

input Copy

```
1 1 2 10
2
1 1 3
2 6 10
```

output Copy

```
-1
```

```cpp
/// Codeforces 242C - King's Path
#include<bits/stdc++.h>
using namespace std;
#define mp make_pair
#define pii pair<int,int>
#define fi first
#define sc second
map< pair<int,int>, int>gg;
map< pair<int,int>, int>vs;
int fx[]={-1,-1,-1,+0,+0,+1,+1,+1};
int fy[]={-1,+0,+1,-1,+1,-1,+0,+1};

int BFS(int sx,int sy,int dx,int dy){
    if(sx==dx&&sy==dy)return 0;

    queue < pair< pair<int,int>, int > > Q;
    Q.push(mp(mp(sx,sy),0));
    vs[mp(sx,sy)]=1;

    while(!Q.empty()){
        auto u = Q.front();
        Q.pop();

        int x = u.fi.fi;
        int y = u.fi.sc;
        int w = u.sc;

        for(int k=0; k<8; k++){
            int xx = x+fx[k];
            int yy = y+fy[k];

            if(gg.find(mp(xx,yy)) == gg.end())continue;
            if(vs.find(mp(xx,yy)) != vs.end())continue;

            if(xx==dx && yy==dy)return w+1;
            vs[mp(xx,yy)]=1;
            Q.push(mp(mp(xx,yy),w+1));
        }
    }
    return -1;
}
int main()
{
    ios::sync_with_stdio(false); cin.tie(0);
    int sx,sy,dx,dy;
    cin>>sx>>sy>>dx>>dy;
    int q; cin>>q;
    while(q--){
        int r,a,b; cin>>r>>a>>b;
        for(int i=a; i<=b; i++){
            gg[mp(r,i)]=1;
        }
    }

    cout << BFS(sx,sy,dx,dy) << endl;

    return 0;
}
```