Description | Hints | Submissions | Discussions | Notes

Burn them All

[illegible] Difficulty | 2 sec Time Limit | 256000KB Memory | 100 Score

80/80 XP | 30/30

Description

You have given an undirected graph of **N** vertices and **M** edges. Edge weight **d** on edge between nodes **u** and **v** represents that **u** and **v** are connected by a thread of length **d** units.

You set node **A** on to the fire. It takes to fire 1 unit of time to travel 1 unit of distance via threads.

Let **T** be the minimum time in which all the threads will be burned out.

Your task is to find **10T**. We can prove that **10T** will always be an integer number.

See the sample test cases for more explanation.

Input Format

The first line of input contains **N –** the number of nodes in the graph.

The second line contains **M** – the number of edges in the graph.

Next **M** lines contain three integers **u**, **v**, **d** – there is a thread between node **u** and **v** of length **d**.

The last line of input contains **A** – the node on which we set fire.

It's guaranteed that graph is connected.

Output Format

Print the value of **10T**.

Constraints

$1 \le N \le 2 \times 10^5$

$1 \le M \le 2 \times 10^5$

$1 \le u, v \le N$

$1 \le d \le 10^9$

Sample Input 1 Copy

C++14 | 00:00:00 | 12 px

```
            if(dis[nei]>dis[T.S]+wt){
                dis[nei]=dis[T.S]+wt;
                pq.push(MP(-dis[nei],nei));
            }
        }
    }
    lli ans=INT_MIN;
    for(auto i:edges){
        lli a=dis[i.F.F];
        lli b=dis[i.F.S];
        lli wt=i.S;
        if(abs(a-b)==wt){
            ans=max(ans,max(a,b)*10);
        }
        else{
            ans=max(ans,(a+b+wt)*5);
        }
    }
    cout<<ans<<endl;
}
int main(){
    ios_base::sync_with_stdio(0);
    cin.tie(0);
```

Sample Tests | Manual Tests

Test Case 1 | Test Case 2 | Test Case 3

ACCEPTED

Input

2

Console | Run on Sample