

# FROM SOURCE TO DESTINATION

\*\*\*\*\*

**Author:** Aashish Goel

\*\*\*\*\*

## Explanation:

Let's note that there is a value for the fuel tank capacity (call it  $w$ ), that if the vehicle has the fuel tank capacity equal or more than  $w$  it will be able to reach the destination B in time, else — will not be able.

The value  $w$  can be found with help of binary search because the function  $\text{isok}(w)$  (it is possible and it has enough time for such cur) is monotonic — in the beginning all values of this function is false, but after some moment the values of this function is always true.

After we found  $w$  it remain only to choose the cheapest vehicle from the vehicles which fuel tank capacity equal or more than  $w$ .

The function  $\text{isok}(w)$  can be realized with greedy algorithm. It is easy to write down the formula for find the number of kilometers which we can ride in fast mode if the nearest gas station is on the distance  $x$  and we have  $f$  liters of fuel in fuel tank:

if  $x > f$ , then it is impossible to reach the nearest gas station and  $\text{isok}(w)$  must return false,

if  $x \leq f$ , then it is possible to ride in the fast mode  $\min(x, f - x)$  kilometers.

So, now we know how to find the value  $\text{isok}(w)$  in one iterate through the array of gas stations in the increasing order of their positions.

## Code:

```
#include<bits/stdc++.h>
#define me(x) memset(x,0,sizeof(x))
#define fo(x) for(LL i=1;i<=x;i++)
using namespace std;
typedef long long LL;
const int maxd=2e6+5;
LL c[maxd],v[maxd],g[maxd];
LL n,k,s,t;
LL isok(LL d){
    LL sum=0;
    fo(k+1){
        if(g[i]-g[i-1]>d){
            return 0;
        }else{
            sum+=2*(g[i]-g[i-1]);
            sum-=min(d-g[i]+g[i-1],g[i]-g[i-1]);
        }
    }
    return sum<=t;
}

int main(){
    ios::sync_with_stdio(false);
    cin.tie(0);
    while(cin>>n>>k>>s>>t){
```

```

    for(LL i=1;i<=n;i++){
        cin>>c[i]>>v[i];
    }for(LL i=1;i<=k;i++){
        cin>>g[i];
    }g[0]=0;g[k+1]=s;
    sort(g,g+k+2);
    LL l=1,r=1e10,mid;
    while(l<=r){
        mid=(l+r)/2;
        if(isok(mid)) r=mid-1;
        else l=mid+1;
    }
    LL ans=0x3f3f3f3f;
    for(LL i=1;i<=n;i++){
        if(v[i]>=l){
            ans=min(ans,c[i]);
        }
    }if(ans==0x3f3f3f3f){
        cout<<"-1\n";
    }else{
        cout<<ans<<endl;
    }
}
return 0;
}

```