Convex Hull

https://dmoj.ca/problem/ccc15s4

Problem

There is a graph of N ($2 \leq N \leq 2{,}000$) vertices with M ($1 \leq M \leq 10{,}000$) weighted undirected edges. Each edge also has a certain second weight. What is the length shortest path between two given points A and B, if the sum of the second weights of each edge on the path must sum to below a given value K?

Solution

The solution to this question is quite similar to that of CrossCountry Canada; by representing the graph as a set of N*K vertices, a new graph can be constructed in which the base Dijkstra's algorithm can be used.

```
#include <bits/stdc++.h>
using namespace std;

#define f first
#define s second

int K,N,M,A,B,dist[2010][210];
vector<pair<int,pair<int,int>>> G[2010];
priority_queue<pair<int, pair<int, int>>, vector<pair<int, pair<int,
int>>>, greater<pair<int, pair<int, int>>>> Q;

int main(){
    cin >> K >> N >> M;
    for(int i = 0; i < M; i++){
        int a,b,t,h;
        cin >> a >> b >> t >> h;
        G[a].push_back({b,{t,h}});
        G[b].push_back({a,{t,h}});
    }
    memset(dist,-1,sizeof dist);
    cin >> A >> B;

    Q.push({0,{A,0}});

    while(!Q.empty()){
        pair<int, pair<int, int>> u = Q.top();
        Q.pop();

        if(dist[u.s.f][u.s.s] == -1){
            dist[u.s.f][u.s.s] = u.f;
            for(pair<int, pair<int, int>> i : G[u.s.f]){
                if(u.s.s + i.s.s < K) Q.push({u.f + i.s.f, {i.f,u.s.s
+ i.s.s}});
            }
        }
    }
    int tot = 1e+9;
    for(int i = 0; i < K; i++) if(dist[B][i] != -1) tot = min(tot,
dist[B][i]);
    if(tot == 1e+9) cout << -1 << endl;
    else cout << tot << endl;
}
```