

```

1  #include <iostream>
2  #include <algorithm>
3  #include <iomanip>
4  #include <map>
5  #include <set>
6  #include <queue>
7  #include <stack>
8  #include <numeric>
9  #include <bitset>
10 #include <cmath>
11
12 static const int MOD = 1000000007;
13 using ll = long long;
14 using u32 = uint32_t;
15 using namespace std;
16
17 template<class T> constexpr T INF = ::numeric_limits<T>::max()/32*15+208;
18
19 int main() {
20
21     return 0;
22 }
23
24 template <typename T>
25 struct edge {
26     int from, to; T cost;
27     edge(int to, T cost) : from(-1), to(to), cost(cost) {}
28     edge(int from, int to, T cost) : from(from), to(to), cost(cost) {}
29 };
30
31 template <typename T>
32 vector<T> dijkstra(int s, vector<vector<edge<T>>> &G) {
33     size_t n=G.size();
34     vector<T> d(n, INF<T>);
35     priority_queue<pair<T, int>, vector<pair<T, int>>, greater<>> Q;
36     d[s]=0;
37     Q.emplace(0,s);
38     while(!Q.empty()) {
39         T cost; int i;
40         tie(cost, i) = Q.top(); Q.pop();
41         if(d[i] < cost) continue;
42         for (auto &e : G[i]) {
43             auto cost2 = cost + e.cost;
44             if(d[e.to] <= cost2) continue;
45             d[e.to] = cost2;
46             Q.emplace(d[e.to], e.to);
47         }
48     }
49     return d;
50 }
51
52 template<typename T>
53 T extgcd(T a, T b, T &x ,T &y){
54     for (T u = y = 1, v = x = 0; a; ) {

```

```
4         ll q = b/a;
5         swap(x -= q*u, u);
6         swap(y -= q*v, v);
7         swap(b -= q*a, a);
8     }
9     return b;
10 }
11
```