```
#include <bits/stdc++.h>
using namespace std;
#define ll long long
#define inf (int)1e9
#define node tuple<int , int>

```

Dijkstra:

```
vector<int> D(100100 , inf);
vector<pair<int , int>> adj[100100];

void dijkstra (int src){
    priority_queue <node , vector<node> , greater<node>> q;
    D[src] = 0;
    q.push({0 , src});
    while(!q.empty()){
        auto [d ,u] = q.top();
        q.pop();
        if(d > D[u]) continue;
        for(auto[v , w] : adj[u]){
            if(D[v] > D[u]+w){
                // parent[v] = u; // if you need to print path
                D[v] = D[u]+w;
                q.push({D[v] , v});
            }
        }
    }
}

// to print the path:
vector<int> parent(100100 , -1);

void printPath(int j){
    if (parent[j] == - 1){
        cout<<j<<" ";
        return;
    }
    printPath(parent[j]);
    cout<<j<<" ";
}

```

Floyd:

```
vector<vector<int>> dist(1000 , vector<int> (1000));

void floyd(){
    for(int k = 1 ; k <= 100 ; k++) //check n
        for(int i = 1 ; i <= 100 ; i++) //check n
            for(int j = 1 ; j <=100 ; j++) //check n
                dist[i][j] = min(dist[i][j] , dist[i][k]+dist[k][j]);
}
```