1. Single source sortest path

A. Dijkastra Algo (not for -ive circle)
// Bellman Ford Algo (for negative circle)

```
#include<bits/stdc++.h>
using namespace std;
vector<pair<int,int>>adj[20000007];
void addEdge(int u, int v, int w)
{
        adj[u].push_back({v, w});
        adj[v].push_back({u, w});
}

void primMST(int V)
{
priority_queue<pair<int, int>, vector <pair<int, int>> , greater<pair<int, int>> > pq;
    int src = 0;
        vector<int>dis(V, INT_MAX);
        vector<bool> inMST(V, false);
        pq.push({0, src});
        dis[src] = 0;
       while (!pq.empty())
        {
                int u = pq.top().second;
                pq.pop();

       if(inMST[u] == true)
                continue;
                inMST[u] = true;

       for(auto it:adj[u])
                {
                        int v=it.first;
                        int weight=it.second;

                        if (inMST[v] == false && dis[v] > dis[u] + weight)
                        {
                                dis[v] = dis[u] + weight;
                                pq.push({dis[v], v});
```

```
                        }
                    }
            }
            for(int i=0;i<V;i++)
            cout<<dis[i]<<"  ";
}
int main()
{

            int V = 9;
            addEdge(0, 1, 4);
            addEdge(0, 7, 8);
            addEdge(1, 2, 8);
            addEdge(1, 7, 11);
            addEdge(2, 3, 7);
            addEdge(2, 8, 2);
            addEdge(2, 5, 4);
            addEdge(3, 4, 9);
            addEdge(3, 5, 14);
            addEdge(4, 5, 10);
            addEdge(5, 6, 2);
            addEdge(6, 7, 1);
            addEdge(6, 8, 6);
            addEdge(7, 8, 7);

    primMST(V);
    return 0;
}
```