```cpp
#include "../../bits/stdc++.h"
#include "../graph.hpp"

// 单一始点最短経路, O(ElogV)
std::vector<int> dijkstra(const Graph &g, int s)
{
    std::vector<int> d((int)g.size(), INF);
    d[s] = 0;
    std::priority_queue<P, std::vector<P>, std::greater<P>> que;
    que.push(P(0, s)); // P(dis, pos)
    while (!que.empty())
    {
        int dist = que.top().first;
        int v = que.top().second;
        que.pop();
        if (d[v] < dist)
            continue;
        for (const auto &e : g[v])
        {
            if (d[e.to] <= d[v] + e.cost)
                continue;
            d[e.to] = d[v] + e.cost;
            que.push(P(d[e.to], e.to));
        }
    }
    return d;
}

```