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

// verified: https://atcoder.jp/contests/abc126/submissions/5485170
std::vector<int> bfs01(const Graph &g, int s)
{
    int n = g.size();
    std::vector<int> d(n, INF);
    std::deque<int> que;
    que.push_back(s);
    d[s] = 0;
    while (!que.empty())
    {
        int cur = que.front();
        que.pop_front();
        for (auto p : g[cur])
        {
            int dst = p.to, w = p.cost;
            assert(w == 0 || w == 1);
            if (d[dst] != INF)
                continue;
            if (w == 0)
            {
                d[dst] = d[cur];
                que.push_front(dst);
            }
            else
            {
                d[dst] = d[cur] + 1;
                que.push_back(dst);
            }
        }
    }
    return d;
}

```