```cpp
#include "../bits/stdc++.h"
// Edge, Graph
#include "./graph.hpp"
/**
 * 全域最小カット(最大は NP-hard...)
 * Stort-Wagner/Nagamochi-Ibaraki
 * O(|V|^3) (最大隣接順序にフィボナッチヒープを使うと O(|E||V|+|V|log|V|))
 */
// verify 用問題あったらください
int minimumCut(const Graph &g)
{
    int n = g.size();
    std::vector<std::vector<int>> h(n, std::vector<int>(n)); // 隣接行列
    for (int i = 0; i < n; i++)
    {
        for (const auto &e : g[i])
        {
            h[e.from][e.to] += e.cost;
        }
    }
    std::vector<int> V(n);
    for (int i = 0; i < n; i++)
        V[i] = i;

    int ret = 1e9;
    for (int i = n; i > 1; i--)
    {
        std::vector<int> ws(i, 0);
        int u, v;
        int w;
        for (int j = 0; j < i; j++)
        {
            u = v;
            v = std::max_element(ws.begin(), ws.end()) - ws.begin();
            w = ws[v];
            ws[v] = -1;
            for (int k = 0; k < i; k++)
                if (ws[k] >= 0)
                    ws[k] += h[V[v]][V[k]];
        }
        for (int j = 0; j < i; j++)
        {
            h[V[j]][V[u]] += h[V[j]][V[v]];
            h[V[u]][V[j]] += h[V[v]][V[j]];
        }
        V.erase(V.begin() + v);
        ret = std::min(ret, w);
    }
    return ret;
}

```