```cpp
const ll LINF = 0x3f3f3f3f3f3f3f3fLL;
const int MAXN = 3050;

struct edge {
    // flow in the edge = orcap - cap
    int viz, cap, orcap, cost, dual;
    edge(int viz, int cap, int cost, int dual) : viz(viz), cap(cap), orcap(cap), cost(cost), dual(dual) {}
};

vector<edge> g[MAXN];
ll d[MAXN];
int p[MAXN], p edge[MAXN], color[MAXN];
ll flow, fcost;

void add edge(int x, int y, int cap, int cost) {
    g[x].pb(edge(y, cap, cost, (int)g[y].size()));
    g[y].pb(edge(x, 0, -cost, (int)g[x].size() - 1));
}

int SPFA(int s, int t, int n) {
    const int WHITE = 0, GRAY = 1;
    int next, viz, cap, cost;
    queue<int> fila;
    FOR0(i,n) {
        d[i] = LINF;
        color[i] = WHITE;
    }
    d[s] = 0;
    fila.push(s);
    while (!fila.empty()) {
        next = fila.front();
        fila.pop();
        color[next] = WHITE;
        FOR0(i,sz(g[next])) {
            viz = g[next][i].viz;
            cost = g[next][i].cost;
            cap = g[next][i].cap;
            if (cap && d[viz] > d[next] + cost) {
                d[viz] = d[next] + cost;
                p[viz] = next;
                p edge[viz] = i;
                if (color[viz] == WHITE) {
                    color[viz] = GRAY;
                    fila.push(viz);
                }
            }
        }
    }
    return d[t] != LINF;
}

void mcmf(int s, int t, int n) {
    ll augment;
    int idx, dual;
    flow = 0;
    fcost = 0;
    while (SPFA(s, t, n)) {
        augment = LINF;
        for (int v = t; v != s; v = p[v]) {
            idx = p edge[v];
            augment = min(augment, 1LL*g[p[v]][idx].cap);
        }
        for (int v = t; v != s; v = p[v]) {
            idx = p edge[v];
            dual = g[p[v]][idx].dual;
            g[p[v]][idx].cap -= augment;
            g[v][dual].cap += augment;
        }
        flow += augment;
```

```
70            fcost += augment*d[t];
71        }
72    }
73
```