

```

#include <bits/stdc++.h>

#define int long long int

#define inf 1e16

using namespace std;

vector<vector<pair<int, int> >> g;

int32_t main()
{
    int source, dest;

    cin >> source >> dest;

    int n, m;

    cin >> n >> m;

    vector<int> greenLight(n + 1);

    g = vector<vector<pair<int, int> >> (n + 1);

    for (int i = 1; i <= n; ++i)
        cin >> greenLight[i];

    for (int i = 0; i < m; ++i) {
        int u, v, w;

        cin >> u >> v >> w;

        g[u].push_back({ v, w });

        g[v].push_back({ u, w });
    }

    priority_queue<pair<int, int>, vector< pair<int, int> >, greater<pair<int, int>>> q;

    q.push({ source, 0 });

    vector<int> dist(n + 1, inf);

    vector<int> vis(n + 1, 0);

    dist[source] = 0;

    while (!q.empty()) {
        pair<int, int> p = q.top();

```

```

q.pop();

int u = p.first;

vis[u] = 1;

for (auto p: g[u]) {
    int v = p.first;
    int w = p.second;
    if (v == dest) {
        int Dist = dist[u] + w;
        if (dist[v] > Dist) {
            dist[v] = Dist;
            q.push({ v, dist[v] });
        }
    }
    else {
        int Dist = dist[u] + w;
        int extra = (Dist % greenLight[v] == 0 ? 0 : greenLight[v] - (Dist % greenLight[v]));
        if (dist[v] > Dist + extra) {
            dist[v] = Dist + extra;
            q.push({ v, dist[v] });
        }
    }
}

cout << dist[dest] << '\n';

return 0;

}

```