

# ICPC Notebook

template	
1.sh	1
data-structure	
ST表.hpp	1
单调栈.hpp	1
单调队列.hpp	2
树状数组-双区间.hpp	2
树状数组1.hpp	2
树状数组2.hpp	3
线段树1.hpp	3
线段树2.hpp	4
math	
modint	
FPS	
graph	
dijkstra-负权.hpp	5
dijkstra.hpp	5
floyd.hpp	6
kruscal.hpp	6
prim.hpp	6
tarjan强连通分量.hpp	7
并查集.hpp	8
拓扑排序.hpp	8
缩点.hpp	8
负环-spfa.hpp	9
graph/tree	
LCA-dfn.hpp	9
LCA-tarjan.hpp	10
LCA-倍增.hpp	10
树链剖分.hpp	11
flow	
string	
trie.hpp	12
字符串哈希.hpp	13
algorithm	
geometry	
memo	

template

## 1.sh

XUU's template

data-structure

ST表.hpp

md5: 4be317

```
#include <bits/stdc++.h>
using namespace std;
int lg[100010], f[100010][20], n, m;
int main() {
    ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
    cin >> n >> m;
    for(int i = 2; i <= n; i++) lg[i] = lg[i >> 1] + 1;
    for(int i = 1; i <= n; i++) cin >> f[i][0];
    for(int j = 1; j <= lg[n]; j++) {
        for(int i = 1; i <= n - (1 << j) + 1; i++) { f[i][j] = max(f[i][j - 1], f[i + (1 <<
(j - 1))][j - 1]); }
    }
    for(int i = 1; i <= m; i++) {
        int l, r;
        cin >> l >> r;
        int len = lg[r - l + 1];
        cout << max(f[l][len], f[r - (1 << len) + 1][len]) << "\n";
    }
    return 0;
}
```

单调栈.hpp

md5: f4f08f

```
#include <bits/stdc++.h>
using namespace std;
stack<int> s;
int main() {
    ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
    int n;
    cin >> n;
    int a[n + 1], ans[n + 1], tot = 0;
    for(int i = 1; i <= n; i++) cin >> a[i];
    for(int i = n; i >= 1; i--) {
        int j;
        while(s.size() >= 1) {
            j = s.top();
            if(a[i] < a[j]) {
                ans[++tot] = s.top();
                s.push(i);
                break;
            } else {
                s.pop();
            }
        }
    }
}
```

```
        if(s.empty() == 1) {
            s.push(i);
            ans[++tot] = 0;
        }
    }
    for(int i = tot; i >= 1; i--) cout << ans[i] << " ";
    return 0;
}
```

单调队列.hpp

md5: 2a35c3

```
#include <bits/stdc++.h>
using namespace std;
struct qwq {
    int x, y;
} a;
deque<qwq> q1;
deque<qwq> q2;
int ansmax[10000005];
int ansmin[10000005];
int main() {
    int n, k, w, t = 1;
    cin >> n >> k;
    for(int i = 1; i <= n; i++) {
        cin >> w;
        a.x = w;
        a.y = i;
        while(q1.empty() == 0 && w >= q1.back().x) { q1.pop_back(); }
        while(q2.empty() == 0 && w <= q2.back().x) { q2.pop_back(); }
        q1.push_back(a);
        q2.push_back(a);
        while(i - k >= q1.front().y) { q1.pop_front(); }
        while(i - k >= q2.front().y) { q2.pop_front(); }
        if(i >= k) {
            ansmin[t] = q2.front().x;
            ansmax[t] = q1.front().x;
            t++;
        }
    }
    for(int i = 1; i < t; i++) { cout << ansmin[i] << " "; }
    cout << endl;
    for(int i = 1; i < t; i++) { cout << ansmax[i] << " "; }
    return 0;
}
```

树状数组-双区间.hpp

md5: f893e9

```
#include <bits/stdc++.h>
#define N 500050
#define ll long long
using namespace std;
const ll inf = 2147483647;
ll n, m, t1[N], t2[N];
ll lowbit(int x) { return x & -x; }
void add(ll x, ll k) {
    ll s = x;
    while(x <= n) {
        t1[x] += k;
        t2[x] += s * k;
        x += lowbit(x);
    }
}
```

```
    }
}
ll sum(ll x) {
    ll ans = 0, s = x;
    while(x) {
        ans += (s + 1) * t1[x] - t2[x];
        x -= lowbit(x);
    }
    return ans;
}
int main() {
    ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
    cin >> n >> m;
    for(int i = 1; i <= n; i++) {
        ll x;
        cin >> x;
        add(i, x);
        add(i + 1, -x);
    }
    for(int i = 1; i <= m; i++) {
        ll pd, k, l, r;
        cin >> pd;
        if(pd == 1) {
            cin >> l >> r >> k;
            add(l, k);
            add(r + 1, -k);
        }
        if(pd == 2) {
            cin >> k;
            add(1, k);
            add(2, -k);
        }
        if(pd == 3) {
            cin >> k;
            add(1, -k);
            add(2, k);
        }
        if(pd == 4) {
            cin >> l >> r;
            cout << sum(r) - sum(l - 1) << "\n";
        }
        if(pd == 5) { cout << sum(1) << "\n"; }
    }
    return 0;
}
```

树状数组1.hpp

md5: 726feb

```
#include <bits/stdc++.h>
#define N 500050
#define ll long long
using namespace std;
const ll inf = 2147483647;
int n, m, t[N];
int lowbit(int x) { return x & -x; }
void add(int x, int k) {
    while(x <= n) {
        t[x] += k;
        x += lowbit(x);
    }
}
```

```
    }
}
int sum(int x) {
    int ans = 0;
    while(x) {
        ans += t[x];
        x -= lowbit(x);
    }
    return ans;
}
int main() {
    ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
    cin >> n >> m;
    for(int i = 1; i <= n; i++) {
        int x;
        cin >> x;
        add(i, x);
    }
    for(int i = 1; i <= m; i++) {
        int a, b, c;
        cin >> a >> b >> c;
        if(a == 1) add(b, c);
        if(a == 2) cout << sum(c) - sum(b - 1) << "\n";
    }
    return 0;
}
```

树状数组2.hpp

md5: fcd358

```
#include <bits/stdc++.h>
#define N 500050
#define ll long long
using namespace std;
const ll inf = 2147483647;
int n, m, t[N];
int lowbit(int x) { return x & -x; }
void add(int x, int k) {
    while(x <= n) {
        t[x] += k;
        x += lowbit(x);
    }
}
int sum(int x) {
    int ans = 0;
    while(x) {
        ans += t[x];
        x -= lowbit(x);
    }
    return ans;
}
int main() {
    ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
    cin >> n >> m;
    for(int i = 1; i <= n; i++) {
        int x;
        cin >> x;
        add(i, x);
    }
}
```

```
        add(i + 1, -x);
    }
    for(int i = 1; i <= m; i++) {
        int a, b, c;
        cin >> a;
        if(a == 1) {
            int x, y, z;
            cin >> x >> y >> z;
            add(x, z);
            add(y + 1, -z);
        }
        if(a == 2) {
            int x;
            cin >> x;
            cout << sum(x) << "\n";
        }
    }
    return 0;
}
```

线段树1.hpp

md5: 70febb

```
#include <bits/stdc++.h>
#define ll long long
#define MAXN 100005
using namespace std;
ll n, m, a[MAXN], ans[MAXN << 2], tag[MAXN << 2];
ll ls(ll p) { return p << 1; }
ll rs(ll p) { return p << 1 | 1; }
void f(ll l, ll r, ll p, ll k) {
    ans[p] += (r - l + 1) * k;
    tag[p] += k;
}
void push_up(ll p) { ans[p] = ans[ls(p)] + ans[rs(p)]; }
void push_down(ll l, ll r, ll p) {
    ll mid = (l + r) >> 1;
    f(l, mid, ls(p), tag[p]);
    f(mid + 1, r, rs(p), tag[p]);
    tag[p] = 0;
}
void build(ll l, ll r, ll p) {
    if(l == r) {
        ans[p] = a[l];
        return;
    }
    ll mid = (l + r) >> 1;
    build(l, mid, ls(p));
    build(mid + 1, r, rs(p));
    push_up(p);
}
void update(ll l, ll r, ll p, ll L, ll R, ll k) {
    if(L <= l && r <= R) {
        f(l, r, p, k);
        return;
    }
    push_down(l, r, p);
    ll mid = (l + r) >> 1;
    if(L <= mid) update(l, mid, ls(p), L, R, k);
    if(R > mid) update(mid + 1, r, rs(p), L, R, k);
    push_up(p);
}
```

```
ll query(ll l, ll r, ll p, ll L, ll R) {
    ll res = 0;
    if(L <= l && r <= R) return ans[p];
    ll mid = (l + r) >> 1;
    push_down(l, r, p);
    if(L <= mid) res += query(l, mid, ls(p), L, R);
    if(R > mid) res += query(mid + 1, r, rs(p), L, R);
    return res;
}

int main() {
    ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
    cin >> n >> m;
    for(int i = 1; i <= n; i++) cin >> a[i];
    build(1, n, 1);
    while(m--) {
        int pd, x, y, k;
        cin >> pd;
        if(pd == 1) {
            cin >> x >> y >> k;
            update(1, n, 1, x, y, k);
        }
        if(pd == 2) {
            cin >> x >> y;
            cout << query(1, n, 1, x, y) << endl;
        }
    }
    return 0;
}
```

## 线段树2.hpp

md5: 92efb4

```
#include <bits/stdc++.h>
#define ll long long
#define MAXN 100005
using namespace std;
ll a[MAXN], ans[MAXN << 2], tag_add[MAXN << 2], tag_mul[MAXN << 2], n, q, m;
ll ls(ll p) { return p << 1; }
ll rs(ll p) { return p << 1 | 1; }
void f(ll l, ll r, ll p, ll k_mul, ll k_add) {
    ans[p] = (ans[p] * k_mul + k_add * (r - l + 1)) % m;
    tag_mul[p] = (tag_mul[p] * k_mul) % m;
    tag_add[p] = (tag_add[p] * k_mul + k_add) % m;
}
void push_up(ll p) { ans[p] = (ans[ls(p)] + ans[rs(p)]) % m; }
void push_down(ll l, ll r, ll p) {
    ll mid = (l + r) >> 1;
    f(l, mid, ls(p), tag_mul[p], tag_add[p]);
    f(mid + 1, r, rs(p), tag_mul[p], tag_add[p]);
    tag_mul[p] = 1;
    tag_add[p] = 0;
}
void build(ll l, ll r, ll p) {
    tag_mul[p] = 1;
    if(l == r) {
        ans[p] = a[l];
        return;
    }
    ll mid = (l + r) >> 1;
    build(l, mid, ls(p));
```

```
    build(mid + 1, r, rs(p));
    push_up(p);
}
void update_mul(ll l, ll r, ll p, ll L, ll R, ll k) {
    if(L <= l && r <= R) {
        tag_add[p] = (tag_add[p] * k) % m;
        tag_mul[p] = (tag_mul[p] * k) % m;
        ans[p] = (ans[p] * k) % m;
        return;
    }
    push_down(l, r, p);
    ll mid = (l + r) >> 1;
    if(L <= mid) update_mul(l, mid, ls(p), L, R, k);
    if(R > mid) update_mul(mid + 1, r, rs(p), L, R, k);
    push_up(p);
}
void update_add(ll l, ll r, ll p, ll L, ll R, ll k) {
    if(L <= l && r <= R) {
        ans[p] = (ans[p] + (r - l + 1) * k) % m;
        tag_add[p] = (tag_add[p] + k) % m;
        return;
    }
    push_down(l, r, p);
    ll mid = (l + r) >> 1;
    if(L <= mid) update_add(l, mid, ls(p), L, R, k);
    if(R > mid) update_add(mid + 1, r, rs(p), L, R, k);
    push_up(p);
}
ll query(ll l, ll r, ll p, ll L, ll R) {
    ll res = 0;
    if(L <= l && r <= R) return ans[p] % m;
    push_down(l, r, p);
    ll mid = (l + r) >> 1;
    if(L <= mid) res += query(l, mid, ls(p), L, R) % m;
    if(R > mid) res += query(mid + 1, r, rs(p), L, R) % m;
    return res % m;
}
int main() {
    ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
    cin >> n >> q >> m;
    for(int i = 1; i <= n; i++) cin >> a[i];
    build(1, n, 1);
    while(q--) {
        int x, y, k, pd;
        cin >> pd;
        if(pd == 1) {
            cin >> x >> y >> k;
            update_mul(1, n, 1, x, y, k);
        }
        if(pd == 2) {
            cin >> x >> y >> k;
            update_add(1, n, 1, x, y, k);
        }
        if(pd == 3) {
            cin >> x >> y;
            cout << query(1, n, 1, x, y) << "\n";
        }
    }
}
```

<div>return 0;</div>
math
modint
FPS
graph

dijkstra-负权.hpp

md5: dcd40

```
#include <bits/stdc++.h>
#define N 60006
#define ll long long
using namespace std;
const ll inf = 1e9;
struct edge {
    int v, next, w;
} e[N << 1];
struct node {
    ll pos, d;
    bool operator<(const node& x) const { return x.d < d; }
};
int head[N], cnt, n, m, tot[N];
ll h[N], dis[N];
bool vis[N];
void add(int u, int v, int w) {
    e[++cnt].v = v;
    e[cnt].w = w;
    e[cnt].next = head[u];
    head[u] = cnt;
}
bool spfa() {
    queue<int> q;
    memset(h, 0x7f, sizeof(h));
    q.push(0);
    h[0] = 0;
    vis[0] = 1;
    while(!q.empty()) {
        int u = q.front();
        q.pop();
        vis[u] = 0;
        for(int i = head[u]; i; i = e[i].next) {
            int v = e[i].v, w = e[i].w;
            if(h[v] > h[u] + w) {
                h[v] = h[u] + w;
                tot[v] = tot[u] + 1;
                if(tot[v] > n) return 1;
                if(vis[v] == 0) q.push(v), vis[v] = 1;
            }
        }
    }
    return 0;
}
void dijkstra(int s) {
    priority_queue<node> q;
    memset(vis, 0, sizeof(vis));
```

```
for(int i = 1; i <= n; i++) dis[i] = inf;
dis[s] = 0;
q.push({s, 0});
while(!q.empty()) {
    int u = q.top().pos;
    q.pop();
    if(vis[u] == 1) continue;
    vis[u] = 1;
    for(int i = head[u]; i; i = e[i].next) {
        int v = e[i].v, w = e[i].w;
        if(dis[v] > dis[u] + w) {
            dis[v] = dis[u] + w;
            if(vis[v] == 0) q.push({v, dis[v]});
        }
    }
}
}
int main() {
    ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
    cin >> n >> m;
    for(int i = 1; i <= m; i++) {
        int x, y, z;
        cin >> x >> y >> z;
        add(x, y, z);
    }
    for(int i = 1; i <= n; i++) add(0, i, 0);
    if(spfa() == 1) {
        cout << -1;
        return 0;
    }
    for(int u = 1; u <= n; u++)
        for(int i = head[u]; i; i = e[i].next) e[i].w += h[u] - h[e[i].v];
    for(int i = 1; i <= n; i++) {
        dijkstra(i);
        ll ans = 0;
        for(int j = 1; j <= n; j++) {
            if(dis[j] == inf) ans += j * inf;
            else ans += j * (dis[j] + h[j] - h[i]);
        }
        cout << ans << "\n";
    }
    return 0;
}
```

dijkstra.hpp

md5: dd41b7

```
#include <bits/stdc++.h>
#define N 500005
#define ll long long
using namespace std;
const ll inf = 2147483647;
struct edge {
    int v, next, w;
} e[N << 1];
struct node {
    int dis, pos;
    bool operator<(const node& x) const { return x.dis < dis; }
};
int n, m, head[N], cnt, s, dis[N];
```

```
bool vis[N];
priority_queue<node> q;
void add(int u, int v, int w) {
    e[++cnt].v = v;
    e[cnt].w = w;
    e[cnt].next = head[u];
    head[u] = cnt;
}
void dijkstra() {
    dis[s] = 0;
    q.push({0, s});
    while(!q.empty()) {
        int x = q.top().pos;
        q.pop();
        if(vis[x] == 1) continue;
        vis[x] = 1;
        for(int i = head[x]; i; i = e[i].next) {
            int v = e[i].v;
            if(dis[v] > dis[x] + e[i].w) {
                dis[v] = dis[x] + e[i].w;
                if(vis[v] == 0) q.push({dis[v], v});
            }
        }
    }
}
int main() {
    ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
    cin >> n >> m >> s;
    for(int i = 1; i <= n; i++) dis[i] = inf;
    for(int i = 1; i <= m; i++) {
        int x, y, z;
        cin >> x >> y >> z;
        add(x, y, z);
    }
    dijkstra();
    for(int i = 1; i <= n; i++) cout << dis[i] << " ";
    return 0;
}
```

## floyd.hpp

md5: 196fe9

```
#include <bits/stdc++.h>
#define N 100005
#define ll long long
using namespace std;
const ll inf = 2147483647;
ll n, m, f[105][105];
int main() {
    ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
    cin >> n >> m;
    for(int i = 1; i <= n; i++)
        for(int j = 1; j <= n; j++)
            if(i != j) f[i][j] = inf;
    for(int i = 1; i <= m; i++) {
        ll x, y, z;
        cin >> x >> y >> z;
        f[x][y] = min(f[x][y], z);
    }
```

```
        f[y][x] = min(f[y][x], z);
    }
    for(int k = 1; k <= n; k++) {
        for(int i = 1; i <= n; i++) {
            if(i == k || f[i][k] == inf) continue;
            for(int j = 1; j <= n; j++) { f[i][j] = min(f[i][j], f[i][k] + f[k][j]); }
        }
    }
    for(int i = 1; i <= n; i++) {
        for(int j = 1; j <= n; j++) cout << f[i][j] << " ";
        cout << "\n";
    }
    return 0;
}
```

## kruscal.hpp

md5: 31a4b1

```
#include <bits/stdc++.h>
#define N 200005
#define ll long long
using namespace std;
const ll inf = 2147483647;
struct edge {
    int u, v, w;
} e[N << 1];
int n, m, f[N], ans, cnt;
int find(int x) {
    if(f[x] == x) return x;
    return f[x] = find(f[x]);
}
bool cmp(edge a, edge b) { return a.w < b.w; }
void kruscal() {
    for(int i = 1; i <= m; i++) {
        int fu = find(e[i].u), fv = find(e[i].v);
        if(fu == fv) continue;
        f[fu] = fv;
        ans += e[i].w;
        if(++cnt == n - 1) break;
    }
}
int main() {
    ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
    cin >> n >> m;
    for(int i = 1; i <= n; i++) f[i] = i;
    for(int i = 1; i <= m; i++) cin >> e[i].u >> e[i].v >> e[i].w;
    sort(e + 1, e + 1 + m, cmp);
    kruscal();
    if(cnt < n - 1) cout << "orz";
    else cout << ans;
    return 0;
}
```

## prim.hpp

md5: e76df0

```
#include <bits/stdc++.h>
#define N 200005
#define ll long long
using namespace std;
const ll inf = 2147483647;
```

```
struct edge {
    int v, next, w;
    bool operator<(const edge& x) const { return x.w < w; }
} e[N << 1];
int head[N], cnt, n, m, tot, now = 1, ans, dis[N];
bool vis[N];
void add(int u, int v, int w) {
    e[++cnt].v = v;
    e[cnt].w = w;
    e[cnt].next = head[u];
    head[u] = cnt;
}
void prim() {
    for(int i = 2; i <= n; i++) dis[i] = inf;
    for(int i = head[1]; i; i = e[i].next) dis[e[i].v] = min(dis[e[i].v], e[i].w);
    while(++tot < n) {
        int minn = inf;
        vis[now] = 1;
        for(int i = 1; i <= n; i++) {
            if(vis[i] == 0 && minn > dis[i]) {
                minn = dis[i];
                now = i;
            }
        }
        if(minn == inf) break;
        ans += minn;
        for(int i = head[now]; i; i = e[i].next) {
            int v = e[i].v;
            if(dis[v] > e[i].w) dis[v] = e[i].w;
        }
    }
}
int main() {
    ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
    cin >> n >> m;
    for(int i = 1; i <= m; i++) {
        int x, y, z;
        cin >> x >> y >> z;
        add(x, y, z);
        add(y, x, z);
    }
    prim();
    if(tot == n - 1) cout << ans;
    else cout << "orz";
    return 0;
}
```

tarjan强连通分量.hpp

md5: 7ce3e4

```
#include <bits/stdc++.h>
#define inf 2147483647
#define ll long long
#define N 100005
using namespace std;
const int mod = 998244353;
struct edge {
    int v, next;
} e[N << 1];
int n, m, cnt, head[N], num, s[N], len, dfn[N], tot, low[N], scc[N];
```

```
vector<int> ans[N];
bool in[N], f[N];
void add(int u, int v) {
    e[++cnt].v = v;
    e[cnt].next = head[u];
    head[u] = cnt;
}
void dfs(int x) {
    dfn[x] = low[x] = ++tot;
    s[++len] = x;
    in[x] = 1;
    for(int i = head[x]; i; i = e[i].next) {
        int y = e[i].v;
        if(dfn[y] == 0) {
            dfs(y);
            low[x] = min(low[x], low[y]);
        } else if(in[y] == 1) low[x] = min(low[x], low[y]);
    }
    if(dfn[x] == low[x]) {
        ans[++num].push_back(x);
        while(x != s[len]) {
            scc[s[len]] = num;
            in[s[len]] = 0;
            ans[num].push_back(s[len]);
            len--;
        }
        len--;
        in[x] = 0;
        scc[x] = num;
    }
}
void solve() {
    cin >> n >> m;
    for(int i = 1; i <= m; i++) {
        int u, v;
        cin >> u >> v;
        add(u, v);
    }
    for(int i = 1; i <= n; i++)
        if(dfn[i] == 0) dfs(i);
    for(int i = 1; i <= n; i++) sort(ans[i].begin(), ans[i].end());
    cout << num << "\n";
    for(int i = 1; i <= n; i++) {
        if(f[scc[i]] == 1) continue;
        f[scc[i]] = 1;
        for(int j = 0; j < ans[scc[i]].size(); j++) { cout << ans[scc[i]][j] << " "; }
        cout << "\n";
    }
}
signed main() {
    ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
    int T = 1;
    // cin>>T;
    while(T--) solve();
    return 0;
}
```

并查集.hpp

md5: 6b2234

```
#include <bits/stdc++.h>
#define N 2000005
#define ll long long
using namespace std;
const ll inf = 2147483647;
int n, m, f[N];
int find(int x) {
    if(f[x] == x) return x;
    return f[x] = find(f[x]);
}
int main() {
    ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
    cin >> n >> m;
    for(int i = 1; i <= n; i++) f[i] = i;
    while(m--) {
        int z, x, y;
        cin >> z >> x >> y;
        if(z == 1) f[find(y)] = find(x);
        if(z == 2) {
            if(find(x) == find(y)) cout << "Y\n";
            else cout << "N\n";
        }
    }
    return 0;
}
```

拓扑排序.hpp

md5: 6c9ce8

```
#include <bits/stdc++.h>
using namespace std;
queue<int> q;
int deg[1001], cnt, head[1001], n;
struct edge {
    int v, next, u;
} e[2002];
void add(int u, int v) {
    e[++cnt].v = v;
    e[cnt].next = head[u];
    head[u] = cnt;
}
void topo() {
    for(int i = 1; i <= n; i++) {
        if(deg[i] == 0) {
            q.push(i);
            cout << i << " ";
        }
    }
    while(!q.empty()) {
        int x = q.front();
        q.pop();
        for(int i = head[x]; i; i = e[i].next) {
            deg[e[i].v]--;
            if(deg[e[i].v] == 0) {
                q.push(e[i].v);
                cout << e[i].v << " ";
            }
        }
    }
}
```

```
    }
}
}
int main() {
    ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
    cin >> n;
    for(int i = 1; i <= n; i++) {
        int x;
        while(cin >> x) {
            if(x == 0) break;
            deg[x]++;
            add(i, x);
        }
    }
    topo();
    return 0;
}
```

缩点.hpp

md5: 3df608

```
#include <bits/stdc++.h>
#define inf 2147483647
#define ll long long
#define N 100005
using namespace std;
const int mod = 998244353;
struct edge {
    int v, next;
} e[N << 1], newe[N << 1];
int n, m, head[N], newhead[N], cnt, newcnt, num, scc[N], dfn[N], low[N], tot, val[N],
s[N], len, dp[N], a[N];
bool in[N];
void add(int u, int v) {
    e[++cnt].v = v;
    e[cnt].next = head[u];
    head[u] = cnt;
}
void newadd(int u, int v) {
    newe[++newcnt].v = v;
    newe[newcnt].next = newhead[u];
    newhead[u] = newcnt;
}
void dfs(int x) {
    dfn[x] = low[x] = ++tot;
    s[++len] = x;
    in[x] = 1;
    for(int i = head[x]; i; i = e[i].next) {
        int y = e[i].v;
        if(dfn[y] == 0) {
            dfs(y);
            low[x] = min(low[x], low[y]);
        } else if(in[y] == 1) low[x] = min(low[x], low[y]);
    }
    if(dfn[x] == low[x]) {
        num++;
        while(s[len] != x) {
            in[s[len]] = 0;
            scc[s[len]] = num;
            val[num] += a[s[len]];
        }
    }
}
```



```
        len--;
    }
    scc[x] = num;
    val[num] += a[x];
    in[x] = 0;
    len--;
}
}
void solve() {
    cin >> n >> m;
    for(int i = 1; i <= n; i++) cin >> a[i];
    for(int i = 1; i <= m; i++) {
        int u, v;
        cin >> u >> v;
        add(u, v);
    }
    for(int i = 1; i <= n; i++)
        if(dfn[i] == 0) dfs(i);
    set<pair<int, int>> st;
    for(int i = 1; i <= n; i++) {
        for(int j = head[i]; j; j = e[j].next) {
            if(scc[i] != scc[e[j].v]) {
                if(st.count({scc[i], scc[e[j].v]}) == 1) continue;
                st.insert({scc[i], scc[e[j].v]});
                newadd(scc[i], scc[e[j].v]);
            }
        }
    }
    for(int i = 1; i <= num; i++) dp[i] = val[i];
    for(int i = num; i >= 1; i--) {
        for(int j = newhead[i]; j; j = newe[j].next) { dp[newe[j].v] = max(dp[newe[j].v],
dp[i] + val[newe[j].v]); }
    }
    int ans = 0;
    for(int i = 1; i <= num; i++) ans = max(ans, dp[i]);
    cout << ans << "\n";
}
signed main() {
    ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
    int T = 1;
    // cin>>T;
    while(T--) solve();
    return 0;
}
```

## 负环-spfa.hpp

md5: ce6dbb

```
#include <bits/stdc++.h>
#define N 60006
#define ll long long
using namespace std;
const ll inf = 2147483647;
struct edge {
    int v, next, w;
} e[N << 1];
int head[N], cnt, tot[N], dis[N], n, m;
bool vis[N];
queue<int> q;
void add(int u, int v, int w){
```

```
    e[++cnt].v = v;
    e[cnt].w = w;
    e[cnt].next = head[u];
    head[u] = cnt;
}
bool spfa() {
    vis[1] = 1;
    dis[1] = 0;
    q.push(1);
    while(!q.empty()) {
        int u = q.front();
        q.pop();
        vis[u] = 0;
        for(int i = head[u]; i; i = e[i].next) {
            int v = e[i].v;
            if(dis[v] > dis[u] + e[i].w) {
                dis[v] = dis[u] + e[i].w;
                tot[v] = tot[u] + 1;
                if(tot[v] >= n) return 1;
                if(vis[v] == 0) vis[v] = 1, q.push(v);
            }
        }
    }
    return 0;
}
void solve() {
    memset(tot, 0, sizeof(tot));
    memset(head, 0, sizeof(head));
    memset(vis, 0, sizeof(vis));
    memset(e, 0, sizeof(e));
    memset(dis, 0x3f, sizeof(dis));
    cnt = 0;
    cin >> n >> m;
    for(int i = 1; i <= m; i++) {
        int x, y, z;
        cin >> x >> y >> z;
        add(x, y, z);
        if(z >= 0) add(y, x, z);
    }
    if(spfa() == 1) cout << "YES\n";
    else cout << "NO\n";
}
int main() {
    ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
    int t;
    cin >> t;
    while(t--) solve();
    return 0;
}
```

## graph/tree

## LCA-dfn.hpp

md5: 78b45a

```
#include <bits/stdc++.h>
#define N 500005
using namespace std;
int lg[N], mi[22][N], n, m, s, dfn[N], dn, head[N], cnt;
struct edge {
```

```
    int v, next;
} e[N << 1];
int get(int x, int y) { return dfn[x] < dfn[y] ? x : y; }
void add(int u, int v) {
    e[++cnt].v = v;
    e[cnt].next = head[u];
    head[u] = cnt;
}
void dfs(int now, int father) {
    dfn[now] = ++dn;
    mi[0][dfn[now]] = father;
    for(int i = head[now]; i; i = e[i].next)
        if(e[i].v != father) dfs(e[i].v, now);
}
int lca(int x, int y) {
    if(x == y) return x;
    if(dfn[x] > dfn[y]) swap(x, y);
    int deep = lg[dfn[y] - dfn[x] - 1];
    return get(mi[deep][dfn[x] + 1], mi[deep][dfn[y] - (1 << deep) + 1]);
}
int main() {
    ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
    cin >> n >> m >> s;
    for(int i = 1; i < n; i++) {
        int x, y;
        cin >> x >> y;
        add(x, y);
        add(y, x);
    }
    for(int i = 2; i <= n; i++) lg[i] = lg[i >> 1] + 1;
    dfs(s, 0);
    for(int i = 1; i <= lg[n]; i++) {
        for(int j = 1; j <= n - (1 << i) + 1; j++) { mi[i][j] = get(mi[i - 1][j], mi[i - 1][j + (1 << i - 1)]); }
    }
    for(int i = 1; i <= m; i++) {
        int x, y;
        cin >> x >> y;
        cout << lca(x, y) << "\n";
    }
    return 0;
}
```

## LCA-tarjan.hpp

md5: 832c2e

```
#include <bits/stdc++.h>
#define MAXN 500005
using namespace std;
int n, m, s, cnt, vcnt, head[MAXN], vhead[MAXN], f[MAXN], lca[MAXN << 2];
bool vis[MAXN];
struct edge {
    int to, next, vto, vnext;
} e[MAXN << 1];
void add(int u, int v) {
    e[++cnt].to = v;
    e[cnt].next = head[u];
    head[u] = cnt;
}
void vadd(int u, int v) {
```

```
    e[++vcnt].vto = v;
    e[vcnt].vnext = vhead[u];
    vhead[u] = vcnt;
}
int find(int x) {
    if(f[x] == x) return x;
    return f[x] = find(f[x]);
}
void tarjan(int u) {
    vis[u] = 1;
    for(int i = head[u]; i; i = e[i].next) {
        int v = e[i].to;
        if(vis[v] == 1) continue;
        tarjan(v);
        f[v] = u;
    }
    for(int i = vhead[u]; i; i = e[i].vnext) {
        int v = e[i].vto;
        if(vis[v] == 0) continue;
        lca[i] = find(v);
        if(i % 2 == 1) lca[i + 1] = lca[i];
        else lca[i - 1] = lca[i];
    }
}
int main() {
    ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
    cin >> n >> m >> s;
    for(int i = 1; i <= n; i++) f[i] = i;
    for(int i = 1; i < n; i++) {
        int x, y;
        cin >> x >> y;
        add(x, y);
        add(y, x);
    }
    for(int i = 1; i <= m; i++) {
        int x, y;
        cin >> x >> y;
        vadd(x, y);
        vadd(y, x);
    }
    tarjan(s);
    for(int i = 1; i <= m; i++) cout << lca[i * 2] << "\n";
    return 0;
}
```

## LCA-倍增.hpp

md5: 7b6e10

```
#include <bits/stdc++.h>
#define MAXN 500005
using namespace std;
struct edge {
    int v, next;
} e[MAXN << 1];
int dep[MAXN], head[MAXN], f[MAXN][22], lg[MAXN], n, m, s, cnt;
void add(int u, int v) {
    e[++cnt].v = v;
    e[cnt].next = head[u];
    head[u] = cnt;
}
```

```
void dfs(int now, int father) {
    f[now][0] = father;
    dep[now] = dep[father] + 1;
    for(int i = 1; i <= lg[dep[now]]; i++) f[now][i] = f[f[now][i - 1]][i - 1];
    for(int i = head[now]; i; i = e[i].next) {
        if(e[i].v != father) dfs(e[i].v, now);
    }
}

int lca(int x, int y) {
    if(dep[x] < dep[y]) swap(x, y);
    while(dep[x] > dep[y]) x = f[x][lg[dep[x] - dep[y]]];
    if(x == y) return x;
    for(int i = lg[dep[x]]; i >= 0; i--) {
        if(f[x][i] != f[y][i]) {
            x = f[x][i];
            y = f[y][i];
        }
    }
    return f[x][0];
}

int main() {
    ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
    cin >> n >> m >> s;
    for(int i = 1; i < n; i++) {
        int x, y;
        cin >> x >> y;
        add(x, y);
        add(y, x);
    }
    for(int i = 2; i <= n; i++) lg[i] = lg[i >> 1] + 1;
    dfs(s, 0);
    for(int i = 1; i <= m; i++) {
        int x, y;
        cin >> x >> y;
        cout << lca(x, y) << "\n";
    }
    return 0;
}
```

## 树链剖分.hpp

md5: 475d70

```
#include <bits/stdc++.h>
#define N 200005
#define ll long long
using namespace std;
const ll inf = 2147483647;
struct edge {
    int v, next;
} e[N << 1];
int n, m, root, mod, ans[N << 2], tag[N << 2], son[N], id[N], fa[N], cnt, dep[N], siz[N],
top[N], head[N], w[N], wt[N];
int ls(int p) { return p << 1; }
int rs(int p) { return p << 1 | 1; }
void push_up(int p) {
    ans[p] = ans[ls(p)] + ans[rs(p)];
    ans[p] %= mod;
}
void f(int l, int r, int p, int k) {
    ans[p] += (r - l + 1) * k;
```

```
    ans[p] %= mod;
    tag[p] += k;
}
void push_down(int l, int r, int p) {
    int mid = (l + r) >> 1;
    f(l, mid, ls(p), tag[p]);
    f(mid + 1, r, rs(p), tag[p]);
    tag[p] = 0;
}
void update(int l, int r, int p, int L, int R, int k) {
    if(L <= l && r <= R) {
        f(l, r, p, k);
        return;
    }
    int mid = (l + r) >> 1;
    push_down(l, r, p);
    if(L <= mid) update(l, mid, ls(p), L, R, k);
    if(R > mid) update(mid + 1, r, rs(p), L, R, k);
    push_up(p);
}
int query(int l, int r, int p, int L, int R) {
    if(L <= l && r <= R) return ans[p];
    int res = 0, mid = (l + r) >> 1;
    push_down(l, r, p);
    if(L <= mid) res += query(l, mid, ls(p), L, R), res %= mod;
    if(R > mid) res += query(mid + 1, r, rs(p), L, R), res %= mod;
    return res % mod;
}
void build(int l, int r, int p) {
    if(l == r) {
        ans[p] = wt[l] % mod;
        return;
    }
    int mid = (l + r) >> 1;
    build(l, mid, ls(p));
    build(mid + 1, r, rs(p));
    push_up(p);
}
void add(int u, int v) {
    e[++cnt].v = v;
    e[cnt].next = head[u];
    head[u] = cnt;
}
void dfs1(int u, int father) {
    dep[u] = dep[father] + 1;
    fa[u] = father;
    siz[u] = 1;
    int hson = -1;
    for(int i = head[u]; i; i = e[i].next) {
        int v = e[i].v;
        if(v == father) continue;
        dfs1(v, u);
        siz[u] += siz[v];
        if(siz[v] > hson) hson = siz[v], son[u] = v;
    }
}
void dfs2(int u, int topc) {
    id[u] = ++cnt;
    wt[cnt] = w[u];
    top[u] = topc;
    if(son[u] == 0) return;
    dfs2(son[u], topc);
```

```

    for(int i = head[u]; i; i = e[i].next) {
        int v = e[i].v;
        if(v == fa[u] || v == son[u]) continue;
        dfs2(v, v);
    }
}

void update1(int x, int y, int k) {
    k %= mod;
    while(top[x] != top[y]) {
        if(dep[top[x]] < dep[top[y]]) swap(x, y);
        update(1, n, 1, id[top[x]], id[x], k);
        x = fa[top[x]];
    }
    if(dep[x] > dep[y]) swap(x, y);
    update(1, n, 1, id[x], id[y], k);
}

int query1(int x, int y) {
    int res = 0;
    while(top[x] != top[y]) {
        if(dep[top[x]] < dep[top[y]]) swap(x, y);
        res += query(1, n, 1, id[top[x]], id[x]);
        res %= mod;
        x = fa[top[x]];
    }
    if(dep[x] > dep[y]) swap(x, y);
    res += query(1, n, 1, id[x], id[y]);
    return res % mod;
}

void update2(int x, int k) { update(1, n, 1, id[x], id[x] + siz[x] - 1, k); }
int query2(int x) { return query(1, n, 1, id[x], id[x] + siz[x] - 1); }

int main() {
    ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
    cin >> n >> m >> root >> mod;
    for(int i = 1; i <= n; i++) cin >> w[i];
    for(int i = 1; i < n; i++) {
        int x, y;
        cin >> x >> y;
        add(x, y);
        add(y, x);
    }
    cnt = 0;
    dfs1(root, 0);
    dfs2(root, root);
    build(1, n, 1);
    while(m--) {
        int op, x, y, z;
        cin >> op;
        if(op == 1) {
            cin >> x >> y >> z;
            update1(x, y, z);
        }
        if(op == 2) {
            cin >> x >> y;
            cout << query1(x, y) << "\n";
        }
        if(op == 3) {
            cin >> x >> y;
            update2(x, y);
        }
        if(op == 4) {

```

```

        cin >> x;
        cout << query2(x) << "\n";
    }
}

return 0;
}

```

flow

string

trie.hpp

md5: 3708ff

```

#include <bits/stdc++.h>
#define ffor(i, a, b) for(int i = (a); i = (b); i++)
#define roff(i, a, b) for(int i = (a); i = (b); i--)
using namespace std;
typedef long long ll;
const int Maxn = 3000005;
string s;
int n, q, t, tot;
int trie[Maxn][130], cnt[Maxn];
void insert(string s) {
    int len = s.size(), u = 0;
    ffor(i, 0, len - 1) {
        int a = s[i];
        if(!trie[u][a]) trie[u][a] = ++tot;
        u = trie[u][a];
        cnt[u]++;
    }
}

int find(string s) {
    int len = s.size(), u = 0;
    ffor(i, 0, len - 1) {
        int a = s[i];
        if(!trie[u][a]) return 0;
        u = trie[u][a];
    }
    return cnt[u];
}

void solve() {
    cinq;
    ffor(i, 0, tot) ffor(j, 0, 122) trie[i][j] = 0;
    ffor(i, 0, tot) cnt[i] = 0;
    tot = 0;
    ffor(i, 1, n) {
        cins;
        insert(s);
    }
    ffor(i, 1, q) {
        cins;
        coutfind(s) 'n';
    }
}

int main() {
    cint;
    while(t--) solve();
    return 0;
}

```

# 字符串哈希.hpp

md5: ec7dfc

```
#include <bits/stdc++.h>
#define N 2000006
#define ll long long
typedef unsigned long long ull;
#define inf 2147483647
using namespace std;
ull base = 131;
ull a[10010];
char s[10010];
int n, ans = 1, prime = 19260817;
ull mod = 212370440130137957;
ull hash1(char s[]) {
    int len = strlen(s);
    ull ans = 0;
    for(int i = 0; i < len; i++) ans = (ans * base + ull(s[i])) % mod + prime;
    return ans;
}
int main() {
    ios::sync_with_stdio(false);
```

```
cin.tie(0);
cout.tie(0);
cin >> n;
for(int i = 1; i <= n; i++) {
    cin >> s;
    a[i] = hash1(s);
}
sort(a + 1, a + 1 + n);
for(int i = 1; i < n; i++) {
    if(a[i] != a[i + 1]) ans++;
}
cout << ans;
return 0;
}
```

algorithm

geometry

memo