CSES PLAN III

HackMD (https://hackmd.io?utm_source=view-page&utm_medium=logo-nav)

CSES PLAN III

因為我跟 `peienwu` 被揍爛了，於是我們決定寫CSES來增進自己的實力
peienwu CSES補完計畫 (https://hackmd.io/@peienwu/cses#CSES-%E8%A3%9C%E5%AE%8C%E8%A8%88%E7%95%AB)

目錄

Advanced Techniques

Meet in the Middle (https://cses.fi/problemset/task/1628)

折半列舉 二分搜

```
#include <iostream>
#include <array>
#include <unordered_map>
#include <algorithm>
#define int long long
using namespace std;
array<int, 40> T;
array<int, 1 << 20> F, S;
void ju(int s, int t, array<int, 1 << 20> &A){
    for(int i = 0; i < 1 << (t - s); i++){
        for(int j = 0; j < (t - s); j++){
            if(i & (1 << j)){
                A[i] = T[s + j] + A[i ^ (1 << j)];
                break;
            }
        }
    }
}
signed main(){
    int n, x, s, cnt = 0;
    cin >> n >> x;
    for(int i = 0; i < n; i++){
        cin >> T[i];
    }
    ju(0, n / 2, F);
    ju(n / 2, n, S);
    s = 1 << (n - n / 2);
    sort(S.begin(), S.begin() + s);
    for(int f : F){
        if(f) cnt += upper_bound(S.begin(), S.begin() + s, x - f) -
    }
    cnt += upper_bound(S.begin(), S.begin() + s, x) - lower_bound(S
    cout << cnt;
    return 0;
}
```

Hamming Distance (https://cses.fi/problemset/task/2136/)

壓常

```cpp
#include <iostream>
#include <array>
#include <bitset>
using namespace std;
array<int, 20004> S;
signed main(){
    cin.tie(0), cout.tie(0), ios::sync_with_stdio(0);
    int n, k, ans = 30;
    char b;
    cin >> n >> k;
    for(int i = 0; i < n; i++){
        for(int j = 0; j < k; j++){
            cin >> b;
            S[i] <<= 1;
            S[i] |= b ^ '0';
        }
    }
    for(int i = 0; i < n; i++){
        for(int j = i + 1; j < n; j++){
            ans = min(ans, (int)__builtin_popcount(S[i] ^ S[j]));
        }
    }
    cout << ans;
    return 0;
}
```

Beautiful Subgrids (https://cses.fi/problemset/task/2137)

壓常

```cpp
#include <bits/stdc++.h>
#pragma GCC target("popcnt")
#define int long long
using namespace std;
array<bitset<3000>, 3000> G;
signed main(){
    cin.tie(0), cout.tie(0), ios::sync_with_stdio(0);
    int n, ans = 0, tmp;
    cin >> n;
    for(int i = 0; i < n; i++){
        cin >> G[i];
    }
    for(int i = 0; i < n; i++){
        for(int j = i + 1; j < n; j++){
            tmp = (G[i] & G[j]).count();
            ans += tmp * (tmp - 1);
        }
    }
    cout << ans / 2;
    return 0;
}
```

Reachable Nodes (https://cses.fi/problemset/task/2138)

壓常 Topological Sort

```cpp
#include <bits/stdc++.h>
#define pb push_back
#pragma target("popcnt")
using namespace std;
array<int, 50004> in;
array<bitset<50004>, 50004> dp;
array<vector<int>, 50004> G;
stack<int> ord;
void topu(int n){
    int u;
    queue<int> Q;
    for(int i = 1; i <= n; i++){
        if(!in[i]) Q.push(i);
    }
    while(!Q.empty()){
        u = Q.front();
        Q.pop();
        ord.push(u);
        for(int v : G[u]){
            in[v]--;
            if(!in[v]) Q.push(v);
        }
    }
    while(!ord.empty()){
        u = ord.top();
        ord.pop();
        dp[u][u] = 1;
        for(int v : G[u]) dp[u] |= dp[v];
    }
}
signed main(){
    cin.tie(0), cout.tie(0), ios::sync_with_stdio(0);
    int n, m, a, b;
    cin >> n >> m;
    while(m--){
        cin >> a >> b;
        G[a].pb(b);
        in[b]++;
    }
    topu(n);
    for(int i = 1; i <= n; i++){
        cout << dp[i].count() << " ";
    }
    return 0;
}
```

Reachability Queries (https://cses.fi/problemset/task/2143)

壓常 Topological Sort SCC

```
#include <bits/stdc++.h>
#define pb push_back
#pragma target("popcnt")
using namespace std;
int cnt = 0;
bitset<50004> vis;
array<int, 50004> in, scc;
array<vector<int>, 50004> G, R, S;
array<bitset<50004>, 50004> dp;
stack<int> out, ord;
void bfs(int u){
    if(vis[u]) return;
    vis[u] = 1;
    for(int v : R[u]) bfs(v);
    out.push(u);
}
void dfs(int u){
    if(scc[u]) return;
    scc[u] = cnt;
    for(int v : G[u]) dfs(v);
}
void topu(int n){
    int u;
    queue<int> Q;
    for(int i = 1; i <= n; i++){
        if(!in[i]) Q.push(i);
    }
    while(!Q.empty()){
        u = Q.front();
        Q.pop();
        ord.push(u);
        for(int v : S[u]){
            in[v]--;
            if(!in[v]) Q.push(v);
        }
    }
    while(!ord.empty()){
        u = ord.top();
        ord.pop();
        dp[u][u] = 1;
        for(int v : S[u]) dp[u] |= dp[v];
    }
}
signed main(){
    int n, m, q, a, b, u;
    cin >> n >> m >> q;
    while(m--){
        cin >> a >> b;
        G[a].pb(b);
        R[b].pb(a);
    }
    for(int i = 1; i <= n; i++){
        bfs(i);
    }
    while(!out.empty()){
        u = out.top();
        out.pop();
        if(!scc[u]) cnt++;
        dfs(u);
    }
    for(int i = 1; i <= n; i++){
        for(int v : G[i]){
            if(scc[v] == scc[i]) continue;
            in[scc[v]]++;
            S[scc[i]].pb(scc[v]);
        }
    }
    topu(cnt);
    while(q--){
        cin >> a >> b;
        cout <<(dp[scc[a]][scc[b]]? "YES\n" : "NO\n");
    }
    return 0;
}
```

Cut and Paste (https://cses.fi/problemset/task/2072)

Treap

```cpp
#include <bits/stdc++.h>
using namespace std;
struct treap{
    int pri, s;
    char x;
    treap *lc, *rc;
    treap(char c){
        pri = rand();
        lc = rc = nullptr;
        x = c;
        s = 1;
    }
    void pull(){
        s = (lc? lc->s : 0) + (rc? rc->s : 0) + 1;
    }
};
int size(treap *t){
    return t? t->s : 0;
}
treap* merge(treap *a, treap *b){
    if(!a || !b) return (!a? b : a);
    if(a->pri > b->pri){
        a->rc = merge(a->rc, b);
        a->pull();
        return a;
    }
    else{
        b->lc = merge(a, b->lc);
        b->pull();
        return b;
    }
}
void split(treap *t, treap *&a, treap *&b, int k){
    if(!t){
        a = b = nullptr;
        return;
    }
    if(size(t->lc) + 1 <= k){
        a = t;
        split(t->rc, a->rc, b, k - size(t->lc) - 1);
        a->pull();
    }else{
        b = t;
        split(t->lc, a, b->lc, k);
        b->pull();
    }
}
void print(treap *t){
    if(!t) return;
    print(t->lc);
    cout << t->x;
    print(t->rc);
}
```

```
signed main(){
    srand(time(NULL));
    int n, q, l, r;
    char x;
    treap *t = nullptr, *a = nullptr, *b = nullptr, *c = nullptr;
    cin >> n >> q;
    for(int i = 0; i < n; i++){
        cin >> x;
        t = merge(t, new treap(x));
    }
    while(q--){
        cin >> l >> r;
        split(t, a, b, l - 1);
        split(b, b, c, r - l + 1);
        t = merge(merge(a, c), b);
    }
    print(t);
    cout << "\n";
    return 0;
}
```

Substring Reversals (https://cses.fi/problemset/task/2073)

`Treap`

```cpp
#include <bits/stdc++.h>
#define np nullptr
using namespace std;
struct treap{
    char x;
    int pri, s;
    bool rev;
    treap *lc, *rc;
    treap(char c){
        pri = rand();
        s = 1;
        x = c;
        rev = 0;
        lc = rc = np;
    }
    void pull(){
        s = (lc? lc->s : 0) + (rc? rc->s : 0) + 1;
    }
    void push(){
        if(!rev) return;
        swap(lc, rc);
        if(lc) lc->rev ^= 1;
        if(rc) rc->rev ^= 1;
        rev = 0;
    }
};
int size(treap *t){
    return t? t->s : 0;
}
treap* merge(treap *a, treap *b){
    if(!a || !b) return a? a : b;
    if(a->pri > b->pri){
        a->push();
        a->rc = merge(a->rc, b);
        a->pull();
        return a;
    }else{
        b->push();
        b->lc = merge(a, b->lc);
        b->pull();
        return b;
    }
}
void split(treap *t, treap *&a, treap *&b, int k){
    if(!t){
        a = b = np;
        return;
    }
    t->push();
    if(k > size(t->lc)){
        a = t;
        split(t->rc, a->rc, b, k - size(t->lc) - 1);
        a->pull();
    }else{
        b = t;
        split(t->lc, a, b->lc, k);
        b->pull();
    }
}
void print(treap *t){
    if(!t) return;
    t->push();
    print(t->lc);
    cout << t->x;
    print(t->rc);
}
signed main(){
    srand(time(NULL));
    int n, q, l, r;
    char x;
    treap *t = np, *a = np, *b = np, *c = np;
    cin >> n >> q;
    for(int i = 0; i < n; i++){
        cin >> x;
        t = merge(t, new treap(x));
    }
    while(q--){
        cin >> l >> r;
        split(t, a, b, l - 1);
        split(b, b, c, r - l + 1);
        b->rev ^= 1;
        t = merge(merge(a, b), c);
    }
    print(t);
    cout << "\n";
    return 0;
}
```

Reversals and Sums (https://cses.fi/problemset/task/2074)

`Treap`

```
#include <bits/stdc++.h>
#define int long long
#define np nullptr
using namespace std;
struct treap{
    int x, sum, s, pri;
    bool rev;
    treap *lc, *rc;
    treap(int v){
        x = v;
        sum = x;
        s = 1;
        pri = rand();
        rev = 0;
        lc = rc = np;
    }
    void pull(){
        s = (lc? lc->s : 0) + (rc? rc->s : 0) + 1;
        sum = (lc? lc->sum : 0) + (rc? rc->sum : 0) + x;
    }
    void push(){
        if(!rev) return;
        swap(lc, rc);
        if(lc) lc->rev ^= 1;
        if(rc) rc->rev ^= 1;
        rev = 0;
    }
};
int size(treap *t){
    return t? t->s : 0;
}
treap* merge(treap *a, treap *b){
    if(!a || !b) return a? a : b;
    if(a->pri > b->pri){
        a->push();
        a->rc = merge(a->rc, b);
        a->pull();
        return a;
    }else{
        b->push();
        b->lc = merge(a, b->lc);
        b->pull();
        return b;
    }
}
void split(treap *t, treap *&a, treap *&b, int k){
    if(!t){
        a = b = np;
        return;
    }
    t->push();
    if(k > size(t->lc)){
        a = t;
        split(t->rc, a->rc, b, k - size(t->lc) - 1);
        a->pull();
    }else{
        b = t;
        split(t->lc, a, b->lc, k);
        b->pull();
    }
}
signed main(){
    int n, q, p, l, r, x;
    treap *t = np, *a = np, *b = np, *c = np;
    cin >> n >> q;
    for(int i = 0; i < n; i++){
        cin >> x;
        t = merge(t, new treap(x));
    }
    while(q--){
        cin >> p >> l >> r;
        split(t, a, b, l - 1);
        split(b, b, c, r - l + 1);
        if(p == 1) b->rev ^= 1;
        else cout << b->sum << "\n";
        t = merge(merge(a, b), c);
    }
    return 0;
}
```

Necessary Roads (https://cses.fi/problemset/task/2076)

橋 DFS Tree

```cpp
#include <bits/stdc++.h>
#define pb push_back
using namespace std;
struct E{
    int u, v;
};
int cnt = 0;
array<int, 100004> low, P;
array<vector<int>, 100004> G;
vector<E> B;
void dfs(int u, int pre){
    P[u] = low[u] = ++cnt;
    for(int v : G[u]){
        if(v == pre) continue;
        if(P[v]) low[u] = min(low[u], P[v]);
        else{
            dfs(v, u);
            if(low[v] > P[u]) B.pb({u, v});
            low[u] = min(low[u], low[v]);
        }
    }
}
signed main(){
    int n, m, a, b;
    cin >> n >> m;
    while(m--){
        cin >> a >> b;
        G[a].pb(b);
        G[b].pb(a);
    }
    dfs(1, 0);
    cout << B.size() << "\n";
    for(E e : B){
        cout << e.u << " " << e.v << "\n";
    }
    return 0;
}
```

Necessary Cities (https://cses.fi/problemset/task/2077)

膝蓋 DFS Tree

```cpp
#include <bits/stdc++.h>
#define pb push_back
using namespace std;
int cnt = 0;
array<int, 100004> P, low;
array<vector<int>, 100004> G;
vector<int> knee;
void dfs(int u, int pre){
    int c = 0;
    bool ok = 0;
    P[u] = low[u] = ++cnt;
    for(int v : G[u]){
        if(v == pre) continue;
        if(P[v]) low[u] = min(low[u], P[v]);
        else{
            c++;
            dfs(v, u);
            if(low[v] >= P[u]) ok = 1;
            low[u] = min(low[u], low[v]);
        }
    }
    if((ok && u > 1) || (u == 1 && c > 1)) knee.pb(u);
}
signed main(){
    int n, m, a, b;
    cin >> n >> m;
    while(m--){
        cin >> a >> b;
        G[a].pb(b);
        G[b].pb(a);
    }
    dfs(1, 0);
    cout << knee.size() << "\n";
    for(int k : knee) cout << k << " ";
    cout << "\n";
    return 0;
}
```

Eulerian Subgraphs (https://cses.fi/problemset/task/2078)

```cpp
#include <bits/stdc++.h>
#define pb push_back
using namespace std;
const int mod = 1e9 + 7;
int ans = 1;
array<int, 100004> vis;
array<vector<int>, 100004>G;
void dfs(int u, int pre){
    vis[u]++;
    for(int v : G[u]){
        if(v == pre || vis[v] > 1) continue;
        if(vis[v] == 1) ans = ans * 2 % mod;
        else dfs(v, u);
    }
    vis[u]++;
}
signed main(){
    int n, m, a, b;
    cin >> n >> m;
    while(m--){
        cin >> a >> b;
        G[a].pb(b);
        G[b].pb(a);
    }
    for(int i = 1; i <= n; i++){
        if(!vis[i]) dfs(i, 0);
    }
    cout << ans << "\n";
    return 0;
}
```

Monster Game I (https://cses.fi/problemset/task/2084)

斜率優化

```
#include <bits/stdc++.h>
#define int long long
#define double long double
using namespace std;
struct line{
    double a, b;
    line(){}
    line(double x, double y): a(x), b(y){}
    double operator*(double x){
        return a * x + b;
    }
    line operator^(line f){
        double x, y;
        x = (b - f.b) / (f.a - a);
        y = f * x;
        return line(x, y);
    }
};
int l, r;
array<double, 200004> F, S, dp;
array<line, 200004> Q;
void pop(double x){
    while(l < r && Q[l] * x > Q[l + 1] * x) l++;
}
void push(line f){
    while(r > l){
        auto [x, y] = f ^ Q[r - 1];
        if(Q[r] * x >= y) r--;
        else break;
    }
    Q[++r] = f;
}
int DP(int n, double x){
    l = 0, r = -1;
    push({x, 0});
    for(int i = 1; i <= n; i++){
        pop(S[i]);
        dp[i] = Q[l] * S[i];
        push({F[i], dp[i]});
    }
    return (int)dp[n];
}
signed main(){
    int n;
    double x;
    cin >> n >> x;
    for(int i = 1; i <= n; i++) cin >> S[i];
    for(int i = 1; i <= n; i++) cin >> F[i];
    cout << DP(n, x) << "\n";
    return 0;
}
```

Monster Game II (https://cses.fi/problemset/task/2085/)

斜率優化 李超線段樹

```
#include <bits/stdc++.h>
#define int long long
#define mid ((l + r) >> 1)
#define lc (p << 1)
#define rc ((p << 1) | 1)
using namespace std;
struct line{
    int a, b;
    int operator+(int x){
        return a * x + b;
    }
};
const int inf = 1e6;
array<int, 200004> S, F, dp;
array<line, 4000004> seg;
void update(int p, int l, int r, line s){
    if(s + mid < seg[p] + mid) swap(s, seg[p]);
    if(l == r) return;
    if(s.a > seg[p].a) update(lc, l, mid, s);
    else update(rc, mid + 1, r, s);
}
int query(int p, int l, int r, int x){
    if(l == r) return seg[p] + x;
    if(x <= mid) return min(seg[p] + x, query(lc, l, mid, x));
    else return min(seg[p] + x, query(rc, mid + 1, r, x));
}
int DP(int n){
    update(1, 1, inf, {F[0], 0});
    for(int i = 1; i <= n; i++){
        dp[i] = query(1, 1, inf, S[i]);
        update(1, 1, inf,{F[i], dp[i]});
    }
    return dp[n];
}
signed main(){
    int n, x;
    cin >> n >> x;
    for(int i = 1; i < 4000004; i++) seg[i] = {inf, inf};
    for(int i = 1; i <= n; i++) cin >> S[i];
    for(int i = 1; i <= n; i++) cin >> F[i];
    F[0] = x;
    cout << DP(n) << "\n";
    return 0;
}
```

Subarray Squares (https://cses.fi/problemset/task/2086)

分治優化

```cpp
#include <bits/stdc++.h>
#define int long long
using namespace std;
array<int, 3004> X;
array<array<int, 3004>, 3004> dp;
int cost(int l, int r){
    return (X[r] - X[l]) * (X[r] - X[l]);
}
void div(int ql, int qr, int l, int r, int k){
    int t, qm = (ql + qr) >> 1;
    dp[k][qm] = 1e18;
    for(int i = l; i < min(r + 1, qm); i++){
        if(dp[k - 1][i] + cost(i, qm) < dp[k][qm]){
            t = i;
            dp[k][qm] = dp[k - 1][i] + cost(i, qm);
        }
    }
    if(ql == qr) return;
    div(ql, qm, l, t, k);
    div(qm + 1, qr, t, r, k);
}
int DP(int n, int k){
    for(int i = 1; i <= n; i++){
        dp[1][i] = X[i] * X[i];
    }
    for(int i = 2; i <= k; i++){
        div(i, n, i - 1, n, i);
    }
    return dp[k][n];
}
signed main(){
    int n, k;
    cin >> n >> k;
    for(int i = 1; i <= n; i++){
        cin >> X[i];
        X[i] += X[i - 1];
    }
    cout << DP(n, k) << "\n";
    return 0;
}
```

Houses and Schools (https://cses.fi/problemset/task/2087)

分治優化

```
#include <bits/stdc++.h>
#define int long long
using namespace std;
array<int, 3004> C;
array<array<int, 3004>, 3004> dis, cst, turn, dp;
void DIS(int n){
    for(int i = 1; i <= n; i++){
        for(int j = i - 1; j > 0; j--){
            dis[i][j] = (i - j) * C[j] + dis[i][j + 1];
        }
        for(int j = i + 1; j <= n; j++){
            dis[i][j] = (j - i) * C[j] + dis[i][j - 1];
        }
    }
    for(int i = 1; i <= n; i++){
        cst[i][i] = 0;
        turn[i][i] = i;
    }
    for(int k = 1; k < n; k++){
        for(int i = 1, j = i + k; j <= n; i++, j++){
            cst[i][j] = 1e18;
            for(int t = turn[i][j - 1]; t <= turn[i + 1][j]; t++){
                if(dis[t][i] + dis[t][j] < cst[i][j]){
                    cst[i][j] = dis[t][i] + dis[t][j];
                    turn[i][j] = t;
                }
            }
        }
    }
}
void div(int ql, int qr, int l, int r, int k){
    int t, qm = (ql + qr) >> 1;
    dp[k][qm] = 1e18;
    for(int i = l; i < min(r + 1, qm); i++){
        if(dp[k][qm] > dp[k - 1][i] + cst[i + 1][qm]){
            dp[k][qm] = dp[k - 1][i] + cst[i + 1][qm];
            t = i;
        }
    }
    if(ql == qr) return;
    div(ql, qm, l, t, k);
    div(qm + 1, qr, t, r, k);
}
int DP(int n, int k){
    for(int i = 1; i <= n; i++){
        dp[1][i] = cst[1][i];
    }
    for(int i = 2; i <= k; i++){
        div(i, n, i - 1, n, i);
    }
    return dp[k][n];
}
signed main(){
    int n, k;
    cin >> n >> k;
    for(int i = 1; i <= n; i++) cin >> C[i];
    DIS(n);
    cout << DP(n, k) << "\n";
    return 0;
}
```

Knuth Division (https://cses.fi/problemset/task/2088)

`Knuth 優化`

```
#include <bits/stdc++.h>
#define int long long
using namespace std;
array<int, 5004> X;
array<array<int, 5004>, 5004> dp, turn;
int DP(int n){
    for(int i = 1; i <= n; i++){
        dp[i][i] = 0;
        turn[i][i] = i;
    }
    for(int k = 1; k < n; k++){
        for(int i = 1, j = i + k; j <= n; i++, j++){
            dp[i][j] = 1e18;
            for(int t = turn[i][j - 1]; t <= turn[i + 1][j]; t++){
                if(dp[i][t] + dp[t + 1][j] < dp[i][j]){
                    turn[i][j] = t;
                    dp[i][j] = dp[i][t] + dp[t + 1][j];
                }
            }
            dp[i][j] += X[j] - X[i - 1];
        }
    }
    return dp[1][n];
}
signed main(){
    int n;
    cin >> n;
    for(int i = 1; i <= n; i++){
        cin >> X[i];
        X[i] += X[i - 1];
    }
    cout << DP(n) << "\n";
    return 0;
}
```

Apples and Bananas (https://cses.fi/problemset/task/2111)

FFT

```cpp
#include <bits/stdc++.h>
#define int long long
#define cmp complex<double>
#define r real
#define i imag
using namespace std;
const int N = 1 << 19;
const double pi = 3.14159265358979323846264;
array<cmp, 1 << 19> A, B, C, X;
cmp ei(double d){
    return {cos(d), sin(d)};
}
void FFT(array<cmp, 1 << 19> &F){
    int n;
    cmp x;
    for(int i = 0, j = 0; i < N; i++){
        if(i > j) swap(F[i], F[j]);
        for(int k = N >> 1; (j ^= k) < k; k >>= 1);
    }
    for(int k = 2; k <= N; k <<= 1){
        n = k >> 1;
        for(int j = 0; j < N; j += k){
            for(int i = j; i < j + n; i++){
                x = X[(i - j) * N / k] * F[i + n];
                F[i + n] = F[i] - x;
                F[i] += x;
            }
        }
    }
}
int rnd(double x){
    double z = (int)x;
    if(x - z >= 0.5) return (int)z + 1;
    else return (int)z;
}
signed main(){
    int k, n, m, x;
    cin >> k >> n >> m;
    for(int i = 0; i < n; i++){
        cin >> x;
        A[x] += 1;
    }
    for(int i = 0; i < m; i++){
        cin >> x;
        B[x] += 1;
    }
    for(int i = 0; i < N; i++){
        X[i] = ei(2 * pi * i / N);
    }
    FFT(A);
    FFT(B);
    for(int i = 0; i < N; i++){
        C[i] = A[i] * B[i];
        X[i] = conj(X[i]);
    }
    FFT(C);
    for(int i = 2; i <= k << 1; i++){
        cout << rnd(C[i].r() / (double)N) << " ";
    }
    cout << "\n";
    return 0;
}
```

One Bit Positions (https://cses.fi/problemset/task/2112)

FFT

```cpp
#include <bits/stdc++.h>
#define cmp complex<double>
#define r real
#define i imag
#define int long long
using namespace std;
const int N = 1 << 19;
const double pi = 3.141592653589793238462643383279502884;
array<cmp, 1 << 19> A, B, C, X;
cmp ei(double d){
    return {cos(d), sin(d)};
}
void FFT(array<cmp, 1 << 19> &F){
    int n;
    cmp x;
    for(int i = 0, j = 0; i < N; i++){
        if(i > j) swap(F[i], F[j]);
        for(int k = N >> 1; (j ^= k) < k; k >>= 1);
    }
    for(int k = 2; k <= N; k <<= 1){
        n = k >> 1;
        for(int j = 0; j < N; j += k){
            for(int i = j; i < j + n; i++){
                x = X[(i - j) * N / k] * F[i + n];
                F[i + n] = F[i] - x;
                F[i] += x;
            }
        }
    }
}
int rnd(double x){
    double z = (int)x;
    if(x - z >= 0.5) return (int)z + 1;
    else return (int)z;
}
signed main(){
    int n;
    string S;
    cin >> S;
    n = S.size();
    for(int i = 0; i < n; i++){
        A[i] = S[i] ^ '0';
        B[n - i - 1] = A[i];
    }
    for(int i = 0; i < N; i++){
        X[i] = ei(2 * pi * i / N);
    }
    FFT(A);
    FFT(B);
    for(int i = 0; i < N; i++){
        C[i] = A[i] * B[i];
        X[i] = conj(X[i]);
    }
    FFT(C);
    for(int i = n; i < (n << 1) - 1; i++){
        cout << rnd(C[i].r() / (double)N) << " ";
    }
    cout << "\n";
    return 0;
}
```

Signal Processing (https://cses.fi/problemset/task/2113)

FFT

```
#include <bits/stdc++.h>
#define cmp complex<double>
#define r real
#define i imag
#define int long long
using namespace std;
const int N = 1 << 19;
const double pi = 3.14159265358979323846264338327950288419716939937
array<cmp, 1 << 19> A, B, C, X;
cmp ei(double d){
    return {cos(d), sin(d)};
}
void FFT(array<cmp, 1 << 19> &F){
    int n;
    cmp x;
    for(int i = 0, j = 0; i < N; i++){
        if(i > j) swap(F[i], F[j]);
        for(int k = N >> 1; (j ^= k) < k; k >>= 1);
    }
    for(int k = 2; k <= N; k <<= 1){
        n = k >> 1;
        for(int j = 0; j < N; j += k){
            for(int i = j; i < j + n; i++){
                x = X[(i - j) * N / k] * F[i + n];
                F[i + n] = F[i] - x;
                F[i] += x;
            }
        }
    }
}
int rnd(double x){
    double z = (int)x;
    if(x - z >= 0.5) return (int)z + 1;
    else return (int)z;
}
signed main(){
    int n, m;
    cin >> n >> m;
    for(int i = 0; i < n; i++){
        cin >> A[i];
    }
    for(int i = m - 1; i >= 0; i--){
        cin >> B[i];
    }
    for(int i = 0; i < N; i++){
        X[i] = ei(2 * pi * i / N);
    }
    FFT(A);
    FFT(B);
    for(int i = 0; i < N; i++){
        C[i] = A[i] * B[i];
        X[i] = conj(X[i]);
    }
    FFT(C);
    for(int i = 0; i < n + m - 1; i++){
        cout << rnd(C[i].r() / (double)N) << " ";
    }
    cout << "\n";
    return 0;
}
```

New Roads Queries (https://cses.fi/problemset/task/2101)

`LCA`

```
#include <bits/stdc++.h>
#define pb push_back
using namespace std;
struct E{
    int v, t;
};
int cnt = 0;
array<int, 200004> in, out, dsu;
array<array<int, 20>, 200004> A, T;
array<vector<E>, 200004> G;
int find(int u){
    if(dsu[u] == u) return u;
    return dsu[u] = find(dsu[u]);
}
void onion(int a, int b){
    int A = find(a), B = find(b);
    dsu[A] = B;
}
void dfs(int u){
    in[u] = ++cnt;
    for(auto [v, t] : G[u]){
        if(in[v]) continue;
        dfs(v);
        T[v][0] = t;
        A[v][0] = u;
    }
    out[u] = ++cnt;
}
void dabo(int n){
    in[0] = 0, out[0] = 1e9;
    for(int j = 1; j < 18; j++){
        for(int i = 1; i <= n; i++){
            A[i][j] = A[A[i][j - 1]][j - 1];
            T[i][j] = max(T[i][j - 1], T[A[i][j - 1]][j - 1]);
        }
    }
}
int query(int a, int b){
    if(find(a) != find(b)) return -1;
    int ans = 0;
    for(int i = 17; i >= 0; i--){
        if(in[A[a][i]] > in[b] || out[A[a][i]] < out[b]){
            ans = max(ans, T[a][i]);
            a = A[a][i];
        }
    }
    if(in[a] > in[b] || out[a] < out[b]){
        ans = max(ans, T[a][0]);
        a = A[a][0];
    }
    for(int i = 17; i >= 0; i--){
        if(in[A[b][i]] > in[a] || out[A[b][i]] < out[a]){
            ans = max(ans, T[b][i]);
            b = A[b][i];
        }
    }
    if(in[b] > in[a] || out[b] < out[a]){
        ans = max(ans, T[b][0]);
        b = A[b][0];
    }
    return ans;
}
signed main(){
    int n, m, q, a, b;
    cin >> n >> m >> q;
    for(int i = 1; i <= n; i++) dsu[i] = i;
    for(int i = 1; i <= m; i++){
        cin >> a >> b;
        if(find(a) == find(b)) continue;
        G[a].pb({b, i});
        G[b].pb({a, i});
        onion(a, b);
    }
    for(int i = 1; i <= n; i++){
        if(!in[i]) dfs(i);
    }
    dabo(n);
    while(q--){
        cin >> a >> b;
        cout << query(a, b) << "\n";
    }
    return 0;
}
```

Dynamic Connectivity (https://cses.fi/problemset/task/2133)

`Segment Tree`

```
#include <bits/stdc++.h>
#define int long long
#define pb push_back
#define mid ((l + r) >> 1)
using namespace std;
struct edge{
    int u, v, l, r;
};
int n, com;
array<int, 100004> DSU;
vector<pair<int, int>> tmp;
stack<vector<pair<int, int>>> chg;
int ehash(int u, int v){
    return u * 100001 + v;
}
pair<int, int> dehash(int x){
    return {x / 100001, x % 100001};
}
int find(int u){
    tmp.pb({u, DSU[u]});
    if(DSU[u] == u){
        chg.push(tmp);
        tmp.clear();
        return u;
    }
    return DSU[u] = find(DSU[u]);
}
void onion(int u, int v){
    int U = find(u), V = find(v);
    if(U == V) return;
    DSU[V] = U;
    com--;
}
void roll(){
    tmp = chg.top();
    chg.pop();
    for(auto [u, d] : tmp){
        DSU[u] = d;
    }
    tmp.clear();
}
void run(int l, int r, vector<edge> &E){
    int c = com;
    vector<edge> D;
    for(edge e : E){
        if(e.l > r || e.r < l) continue;
        if(e.l <= l && e.r >= r) onion(e.u, e.v);
        else D.pb(e);
    }
    if(l == r) cout << com << " ";
    else{
        run(l, mid, D);
        run(mid + 1, r, D);
    }
    for(edge e : E){
        if(e.l <= l && e.r >= r) roll(), roll();
    }
    com = c;
}
signed main(){
    int m, q, t, a, b, p = 0;
    vector<edge> E;
    map<int, int> M;
    cin >> n >> m >> q;
    com = n;
    for(int i = 1; i <= n; i++) DSU[i] = i;
    while(m--){
        cin >> a >> b;
        if(a > b) swap(a, b);
        M[ehash(a, b)] = 0;
    }
    while(q--){
        cin >> t >> a >> b;
        if(a > b) swap(a, b);
        p++;
        if(t == 1) M[ehash(a, b)] = p;
        else{
            E.pb({a, b, M[ehash(a, b)], p - 1});
            M.erase(ehash(a, b));
        }
    }
    for(auto [e, s] : M){
        auto [u, v] = dehash(e);
        E.pb({u, v, s, p});
    }
    run(0, p, E);
    return 0;
}
```

Parcel Delivery (https://cses.fi/problemset/task/2121)

```
Min Cost Max Flow
```

```
#include <bits/stdc++.h>
#define pb push_back
using namespace std;
struct pipe{
    int u, v, f, c;
};
int cnt = 0;
bitset<504> vis, in;
array<int, 504> cost, pre;
array<vector<int>, 504> G;
array<pipe, 2004> E;
void add(int u, int v, int f, int c){
    G[u].pb(cnt);
    E[cnt++] = {u, v, f, c};
    G[v].pb(cnt);
    E[cnt++] = {v, u, 0, -c};
}
void run(int u){
    if(u == 1) return;
    pipe &e = E[pre[u]], &b = E[pre[u] ^ 1];
    e.f++;
    b.f--;
    run(e.v);
}
void SPFA(int s){
    int u, v;
    queue<int> Q;
    cost[s] = 0;
    Q.push(s);
    while(!Q.empty()){
        u = Q.front();
        Q.pop();
        vis[u] = 1;
        in[u] = 0;
        for(int i : G[u]){
            if(!E[i].f) continue;
            v = E[i].v;
            if(cost[u] + E[i].c < cost[v]){
                pre[v] = i ^ 1;
                cost[v] = cost[u] + E[i].c;
                if(!in[v]){
                    in[v] = 1;
                    Q.push(v);
                }
            }
        }
    }
}
int flow(int n, int k){
    int ans = 0;
    while(k--){
        for(int i = 1; i <= n; i++){
            cost[i] = 1e9;
            vis[i] = 0;
        }
        SPFA(1);
        if(!vis[n]) return -1;
        run(n);
        ans += cost[n];
    }
    return ans;
}
signed main(){
    int n, m, k, a, b, f, c;
    cin >> n >> m >> k;
    while(m--){
        cin >> a >> b >> f >> c;
        add(a, b, f, c);
    }
    cout << flow(n, k) << "\n";
    return 0;
}
```

Task Assignment (https://cses.fi/problemset/task/2129)

`Min Cost Max Flow`

```
#include <bits/stdc++.h>
#define pb push_back
using namespace std;
struct pipe{
    int u, v, f, c;
};
int cnt = 0;
bitset<412> in;
array<int, 412> cost, pre;
array<vector<int>, 412> G;
array<pipe, 81204> E;
void add(int u, int v, int f, int c){
    G[u].pb(cnt);
    E[cnt++] = {u, v, f, c};
    G[v].pb(cnt);
    E[cnt++] = {v, u, 0, -c};
}
void run(int u){
    if(!u) return;
    pipe &e = E[pre[u]], &b = E[pre[u] ^ 1];
    b.f--;
    e.f++;
    run(e.v);
}
void SPFA(int s){
    int u, v;
    queue<int> Q;
    cost[s] = 0;
    Q.push(s);
    while(!Q.empty()){
        u = Q.front();
        Q.pop();
        in[u] = 0;
        for(int i : G[u]){
            if(!E[i].f) continue;
            v = E[i].v;
            if(cost[u] + E[i].c < cost[v]){
                cost[v] = cost[u] + E[i].c;
                pre[v] = i ^ 1;
                if(!in[v]){
                    in[v] = 1;
                    Q.push(v);
                }
            }
        }
    }
}
int flow(int n, int k){
    int ans = 0;
    while(k--){
        for(int i = 0; i <= n; i++) cost[i] = 1e9;
        SPFA(0);
        run(n);
        ans += cost[n];
    }
    return ans;
}
signed main(){
    int n, c;
    cin >> n;
    for(int i = 1; i <= n; i++){
        add(0, i, 1, 0);
        add(200 + i, 404, 1, 0);
    }
    for(int i = 1; i <= n; i++){
        for(int j = 1; j <= n; j++){
            cin >> c;
            add(i, 200 + j, 1, c);
        }
    }
    cout << flow(404, n) << "\n";
    for(int i = 1; i <= n; i++){
        cout << i << " ";
        for(int j : G[i]){
            if(!E[j].f) cout << E[j].v - 200 << "\n";
        }
    }
    return 0;
}
```

Distinct Routes II (https://cses.fi/problemset/task/2130)

`Min Cost Max Flow`

```
#include <bits/stdc++.h>
#define pb push_back
using namespace std;
struct pipe{
    int u, v, f, c;
};
int cnt = 0;
bitset<504> vis, in;
array<int, 504> cost, pre;
array<vector<int>, 504> G;
array<pipe, 2004> E;
vector<int> R;
void add(int u, int v, int f, int c){
    G[u].pb(cnt);
    E[cnt++] = {u, v, f, c};
    G[v].pb(cnt);
    E[cnt++] = {v, u, 0, -c};
}
void run(int u){
    if(u == 1) return;
    pipe &e = E[pre[u]], &b = E[pre[u] ^ 1];
    b.f--;
    e.f++;
    run(e.v);
}
void SPFA(int s){
    int u, v;
    queue<int> Q;
    cost[s] = 0;
    Q.push(s);
    while(!Q.empty()){
        u = Q.front();
        Q.pop();
        vis[u] = 1;
        in[u] = 0;
        for(int i : G[u]){
            if(!E[i].f) continue;
            v = E[i].v;
            if(cost[u] + E[i].c < cost[v]){
                cost[v] = cost[u] + E[i].c;
                pre[v] = i ^ 1;
                if(!in[v]){
                    in[v] = 1;
                    Q.push(v);
                }
            }
        }
    }
}
int flow(int n, int k){
    int ans = 0;
    while(k--){
        for(int i = 1; i <= n; i++){
            cost[i] = 1e9;
            vis[i] = 0;
        }
        SPFA(1);
        if(!vis[n]) return -1;
        run(n);
        ans += cost[n];
    }
    return ans;
}
void dfs(int u){
    R.pb(u);
    for(int i : G[u]){
        if(i % 2 == 0 && !E[i].f){
            dfs(E[i].v);
            E[i].f++;
            return;
        }
    }
}
signed main(){
    int n, m, k, a, b;
    cin >> n >> m >> k;
    while(m--){
        cin >> a >> b;
        add(a, b, 1, 1);
    }
    cout << flow(n, k) << "\n";
    while(k--){
        R.clear();
        dfs(1);
        cout << R.size() << "\n";
        for(int u : R) cout << u << " ";
        cout << "\n";
    }
    return 0;
}
```