CSES PLAN IV

CSES PLAN IV

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

Additional Problems

Shortest Subsequence (https://cses.fi/problemset/task/1087)

`Greedy`

```cpp
#include <bits/stdc++.h>
using namespace std;
array<char, 4> C = {'A', 'C', 'G', 'T'};
array<array<int, 1000004>, 4> nxt;
void build(string &DNA){
    int n = DNA.size();
    for(int i = 0; i < 4; i++) nxt[i][n] = n;
    for(int i = n - 1; i >= 0; i--){
        for(int j = 0; j < 4; j++){
            if(DNA[i] == C[j]) nxt[j][i] = i;
            else nxt[j][i] = nxt[j][i + 1];
        }
    }
}
void print(string &DNA){
    int n = DNA.size(), p, now = 0;
    char c;
    for(int i = 0; i < n; i++){
        p = 0;
        for(int j = 0; j < 4; j++){
            if(nxt[j][now] > p){
                p = nxt[j][now];
                c = C[j];
            }
        }
        cout << c;
        now = p + 1;
        if(now > n) return;
   }
}
signed main(){
    string DNA;
    cin >> DNA;
    build(DNA);
    print(DNA);
    return 0;
}
```

Counting Bits (https://cses.fi/problemset/task/1146)

爆搜

```cpp
#include <bits/stdc++.h>
#define int long long
using namespace std;
int cal(int n){
    int ans = 0;
    for(int i = 1ll << 50; i > 0; i >>= 1){
        if(n <= i) continue;
        ans += (n / (i << 1)) * i;
        ans += (n % (i << 1)) - min(i, n % (i << 1));
    }
    return ans;
}
signed main(){
    int n;
    cin >> n;
    cout << cal(n + 1) << "\n";
    return 0;
}
```

Swap Game (https://cses.fi/problemset/task/1670)

BFS

```cpp
#include <bits/stdc++.h>
using namespace std;
struct pos{
    int u, s;
};
int T = 0;
array<int, 10> P;
bitset<387420489> vis;
int swap(int u, int i, int j){
    int a, b;
    a = (u / P[i]) % 9;
    b = (u / P[j]) % 9;
    u += (b - a) * P[i];
    u += (a - b) * P[j];
    return u;
}
int bfs(int S){
    int v;
    queue<pos> Q;
    Q.push({S, 0});
    vis[S] = 1;
    while(1){
        auto [u, s] = Q.front();
        Q.pop();
        if(u == T) return s;
        for(int i = 0; i < 9; i++){
            if(i % 3 < 2){
                v = swap(u, i, i + 1);
                if(!vis[v]){
                    Q.push({v, s + 1});
                    vis[v] = 1;
                }
            }
            if(i + 3 < 9){
                v = swap(u, i, i + 3);
                if(!vis[v]){
                    Q.push({v, s + 1});
                    vis[v] = 1;
                }
            }
        }
    }
}
signed main(){
    int x, G = 0;
    P[0] = 1;
    for(int i = 1; i < 10; i++) P[i] = 9 * P[i - 1];
    for(int i = 0; i < 9; i++) T += i * P[i];
    for(int i = 0; i < 9; i++){
        cin >> x;
        G += (x - 1) * P[i];
    }
    cout << bfs(G) << "\n";
```

```
    return 0;
}
```

Prüfer Code (https://cses.fi/problemset/task/1134)

Priority Queue

```
#include <bits/stdc++.h>
using namespace std;
array<int, 200004> P, L;
void run(int n){
    priority_queue<int, vector<int>, greater<int>> Q;
    for(int i = 1; i <= n; i++){
        if(!L[i]) Q.push(i);
    }
    for(int i = 1; i <= n - 2; i++){
        cout << Q.top() << " " << P[i] << "\n";
        Q.pop();
        if(i == L[P[i]]) Q.push(P[i]);
    }
    cout << Q.top() << " " << P[n - 2] << "\n";
}
signed main(){
    int n;
    cin >> n;
    for(int i = 1; i <= n - 2; i++){
        cin >> P[i];
        L[P[i]] = i;
    }
    L[P[n - 2]] = n;
    run(n);
    return 0;
}
```

Acyclic Graph Edges (https://cses.fi/problemset/task/1756)

DFS Tree

```cpp
#include <bits/stdc++.h>
#define pb push_back
using namespace std;
struct edge{
    int v, d;
};
bitset<100004> vis;
array<int, 100004> dep;
array<vector<edge>, 100004> G;
void dfs(int u, int pre, int d){
    vis[u] = 1;
    dep[u] = d;
    for(auto &[v, t] : G[u]){
        if(vis[v]){
            if(dep[v] > d) t = 1;
        }else{
            dfs(v, u, d + 1);
            t = 1;
        }
    }
}
signed main(){
    int n, m, a, b;
    cin >> n >> m;
    while(m--){
        cin >> a >> b;
        G[a].pb({b, 0});
        G[b].pb({a, 0});
    }
    for(int i = 1; i <= n; i++){
        if(!vis[i]) dfs(i, 0, 1);
    }
    for(int i = 1; i <= n; i++){
        for(auto [v, t] : G[i]){
            if(t) cout << i << " " << v << "\n";
        }
    }
    return 0;
}
```

Strongly Connected Edges (https://cses.fi/problemset/task/2177)

SCC DFS Tree

```cpp
#include <bits/stdc++.h>
#define pb push_back
using namespace std;
struct edge{
    int v, d;
};
int k = 0;
bitset<100004> vis;
array<int, 100004> dep, scc;
array<vector<edge>, 100004> G;
stack<int> out;
void dfst(int u, int pre, int d){
    dep[u] = d;
    for(auto &[v, t] : G[u]){
        if(v == pre) continue;
        if(dep[v]){
            if(dep[v] < d) t = 1;
        }else{
            dfst(v, u, d + 1);
            t = 1;
        }
    }
}
void bfs(int u){
    if(vis[u]) return;
    vis[u] = 1;
    for(auto [v, t] : G[u]){
        if(!t) bfs(v);
    }
    out.push(u);
}
void dfs(int u){
    if(scc[u]) return;
    scc[u] = k;
    for(auto [v, t] : G[u]){
        if(t) dfs(v);
    }
}
signed main(){
    int n, m, a, b, u;
    cin >> n >> m;
    while(m--){
        cin >> a >> b;
        G[a].pb({b, 0});
        G[b].pb({a, 0});
    }
    for(int i = 1; i <= n; i++){
        if(!dep[i]) dfst(i, 0, 1);
    }
    for(int i = 1; i <= n; i++) bfs(i);
    while(!out.empty()){
        u = out.top();
        out.pop();
```

```
54          if(!scc[u]) k++;
55          dfs(u);
56      }
57      if(k > 1) cout << "IMPOSSIBLE\n";
58      else{
59          for(int i = 1; i <= n; i++){
60              for(auto [v, t] : G[i]){
61                  if(t) cout << i << " " << v << "\n";
62              }
63          }
64      }
65      return 0;
66  }
```

Even Outdegree Edges (https://cses.fi/problemset/task/2179)

DFS Tree

```cpp
#include <bits/stdc++.h>
#define pb push_back
using namespace std;
struct edge{
    int v, d;
};
int cnt = 0;
array<bool, 100004> odd;
array<int, 100004> dep;
array<edge, 400004> E;
array<vector<int>, 100004> G;
void add(int a, int b){
    G[a].pb(cnt);
    E[cnt++] = {b, 0};
    G[b].pb(cnt);
    E[cnt++] = {a, 0};
}
void dfs(int u, int pre, int d){
    dep[u] = d;
    for(int i : G[u]){
        auto &[v, t] = E[i];
        if(v == pre) continue;
        if(dep[v]){
            if(dep[v] < d){
                t = 1;
                odd[u] ^= 1;
            }
        }else{
            dfs(v, u, d + 1);
            if(!E[i ^ 1].d){
                t = 1;
                odd[u] ^= 1;
            }
        }
    }
    if(odd[u]){
        for(int i : G[u]){
            auto &[v, t] = E[i];
            if(v == pre){
                t = 1;
                odd[u] ^= 1;
            }
        }
    }
}
signed main(){
    int n, m, a, b;
    cin >> n >> m;
    while(m--){
        cin >> a >> b;
        add(a, b);
    }
    for(int i = 1; i <= n; i++){
```

```
54          if(!dep[i]) dfs(i, 0, 1);
55      }
56      for(int i = 1; i <= n; i++){
57          if(odd[i]){
58              cout << "IMPOSSIBLE\n";
59              return 0;
60          }
61      }
62      for(int i = 1; i <= n; i++){
63          for(int j : G[i]){
64              auto [v, t] = E[j];
65              if(t) cout << i << " " << v << "\n";
66          }
67      }
68      return 0;
69  }
```

Multiplication Table (https://cses.fi/problemset/task/2422)

Binary Search

```
1   #include <bits/stdc++.h>
2   #define int long long
3   using namespace std;
4   int BIS(int n){
5       int l = 0, r = n * n, m = n * n / 2 + (n & 1), mid, s;
6       while(l != r){
7           s = 0;
8           mid = (l + r) >> 1;
9           for(int i = 1; i <= n; i++){
10              s += min(n, mid / i);
11          }
12          if(s < m) l = mid + 1;
13          else r = mid;
14      }
15      return l;
16  }
17  signed main(){
18      int n;
19      cin >> n;
20      cout << BIS(n) << "\n";
21      return 0;
22  }
```

Advertisement (https://cses.fi/problemset/task/1142)

Monoton Stack

```cpp
#include <bits/stdc++.h>
#define int long long
using namespace std;
array<int, 200004> K;
stack<pair<int, int>> F;
int run(int n){
    int ans = 0, p;
    for(int i = 1; i <= n; i++){
        p = i;
        while(!F.empty()){
            auto [l, h] = F.top();
            if(K[i] < h){
                ans = max(ans, h * (i - l));
                F.pop();
                p = l;
            }
            else break;
        }
        F.push({p, K[i]});
    }
    while(!F.empty()){
        auto [l, h] = F.top();
        F.pop();
        ans = max(ans, h * (n - l + 1));
    }
    return ans;
}
signed main(){
    int n;
    cin >> n;
    for(int i = 1; i <= n; i++) cin >> K[i];
    cout << run(n) << "\n";
    return 0;
}
```

Special Substrings (https://cses.fi/problemset/task/2186)

Map

```cpp
#include <bits/stdc++.h>
#define int long long
using namespace std;
array<array<int, 26>, 200004> P;
void build(string &S){
    set<int> M;
    int p = 1;
    for(char s : S){
        for(int i = 0; i < 26; i++){
            P[p][i] = P[p - 1][i];
        }
        P[p++][s - 'a']++;
        M.insert(s - 'a');
    }
    for(int i = 1; i <= S.size(); i++){
        for(int j = 25; j >= 0; j--){
            if(M.find(j) == M.end()) continue;
            P[i][j] -= P[i][0];
        }
    }
}
int run(string &S){
    int ans = 0;
    map<array<int, 26>, int> M;
    for(int i = 0; i <= S.size(); i++){
        ans += M[P[i]];
        M[P[i]]++;
    }
    return ans;
}
signed main(){
    string S;
    cin >> S;
    build(S);
    cout << run(S);
    return 0;
}
```

Permutation Inversions (https://cses.fi/problemset/task/2229)

DP

```cpp
#include <bits/stdc++.h>
#define int long long
using namespace std;
const int mod = 1e9 + 7;
array<array<int, 140004>, 504> dp;
int DP(int n, int k){
    int p, sum;
    dp[1][0] = 1;
    for(int i = 2; i <= n; i++){
        sum = p = 0;
        for(int j = 0; j <= k; j++){
            if(j - p >= i){
                sum -= dp[i - 1][p];
                p++;
            }
            sum += dp[i - 1][j];
            dp[i][j] = sum % mod;
        }
    }
    return dp[n][k];
}
signed main(){
    int n, k;
    cin >> n >> k;
    cout << DP(n, k) << "\n";
    return 0;
}
```

Maximum Xor Subarray (https://cses.fi/problemset/task/1655)

Trie

```cpp
#include <bits/stdc++.h>
using namespace std;
int cnt = 0;
array<int, 200004> X;
array<array<int, 2>, 6000004> trie;
void update(int p, int x, int d){
    if(d < 0) return;
    int c = (x >> d) & 1;
    if(!trie[p][c]) trie[p][c] = ++cnt;
    update(trie[p][c], x, d - 1);
}
int query(int p, int x, int d){
    if(d < 0) return x;
    int c = ((x >> d) & 1) ^ 1;
    if(!trie[p][c]) c ^= 1;
    return query(trie[p][c], x, d - 1) ^ (c << d);
}
int run(int n){
    int x = 0, ans = 0;
    update(0, 0, 30);
    for(int i = 1; i <= n; i++){
        x ^= X[i];
        ans = max(ans, query(0, x, 30));
        update(0, x, 30);
    }
    return ans;
}
signed main(){
    int n;
    cin >> n;
    for(int i = 1; i <= n; i++) cin >> X[i];
    cout << run(n) << "\n";
    return 0;
}
```

Movie Festival Queries (https://cses.fi/problemset/task/1664)

Doubling

```cpp
#include <bits/stdc++.h>
using namespace std;
array<array<int, 24>, 1000004> W;
void build(int n){
    for(int i = 1; i <= n; i++){
        W[i][0] = max(W[i][0], W[i - 1][0]);
    }
    for(int j = 1; j < 20; j++){
        for(int i = 1; i <= n; i++){
            W[i][j] = W[W[i][j - 1]][j - 1];
        }
    }
}
int query(int l, int r){
    int ans = 0;
    for(int i = 19; i >= 0; i--){
        if(W[r][i] >= l){
            ans += 1 << i;
            r = W[r][i];
        }
    }
    return ans;
}
signed main(){
    int n, q, l, r;
    cin >> n >> q;
    while(n--){
        cin >> l >> r;
        W[r][0] = max(l, W[r][0]);
    }
    build(1e6);
    while(q--){
        cin >> l >> r;
        cout << query(l, r) << "\n";
    }
    return 0;
}
```

Chess Tournament (https://cses.fi/problemset/task/1697)

Greedy

```cpp
#include <bits/stdc++.h>
#define pb push_back
using namespace std;
struct ver{
    int u, d;
};
struct cmp{
    bool operator()(ver a, ver b){
        return a.d < b.d;
    }
};
array<int, 100004> deg;
array<vector<int>, 100004> G;
bool run(int n){
    vector<ver> tmp;
    priority_queue<ver, vector<ver>, cmp> Q;
    for(int i = 1; i <= n; i++){
        Q.push({i, deg[i]});
    }
    while(Q.top().d > 0){
        auto[u, d] = Q.top();
        Q.pop();
        while(d--){
            auto [v, t] = Q.top();
            Q.pop();
            if(!t) return 0;
            tmp.pb({v, t - 1});
            G[u].pb(v);
        }
        for(ver v : tmp) Q.push(v);
        tmp.clear();
        Q.push({u, 0});
    }
    return 1;
}
signed main(){
    int n, e = 0;
    cin >> n;
    for(int i = 1; i <= n; i++){
        cin >> deg[i];
        e += deg[i];
    }
    if(!run(n)){
        cout << "IMPOSSIBLE\n";
        return 0;
    }
    cout << e / 2 << "\n";
    for(int i = 1; i <= n; i++){
        for(int v : G[i]) cout << i << " " << v << "\n";
    }
    return 0;
}
```

Tree Traversals (https://cses.fi/problemset/task/1702)

DFS

```cpp
#include <bits/stdc++.h>
using namespace std;
int p = 1;
array<int, 100004> pre, P;
void run(int l, int r){
    int now = pre[p], mid = P[now];
    if(mid > l){
        p++;
        run(l, mid - 1);
    }
    if(mid < r){
        p++;
        run(mid + 1, r);
    }
    cout << now << " ";
}
signed main(){
    int n, in;
    cin >> n;
    for(int i = 1; i <= n; i++) cin >> pre[i];
    for(int i = 1; i <= n; i++){
        cin >> in;
        P[in] = i;
    }
    run(1, n);
    cout << "\n";
    return 0;
}
```

Network Renovation (https://cses.fi/problemset/task/1704)

DFS

```cpp
#include <bits/stdc++.h>
#define pb push_back
using namespace std;
array<vector<int>, 100004> T;
vector<int> leaf;
void dfs(int u, int pre){
    if(T[u].size() == 1) leaf.pb(u);
    for(int v : T[u]){
        if(v == pre) continue;
        dfs(v, u);
    }
}
signed main(){
    int n, a, b;
    cin >> n;
    for(int i = 0; i < n - 1; i++){
        cin >> a >> b;
        T[a].pb(b);
        T[b].pb(a);
    }
    dfs(1, 0);
    if(leaf.size() & 1){
        leaf.pb(leaf[0]);
    }
    cout << leaf.size() / 2 << "\n";
    for(int i = 0; i < leaf.size() / 2; i++){
        cout << leaf[i] << " " << leaf[i + leaf.size() / 2] << "\n"
    }
    return 0;
}
```

Graph Girth (https://cses.fi/problemset/task/1707)

BFS

```cpp
#include <bits/stdc++.h>
#define pb push_back
using namespace std;
array<int, 2504> dis;
array<vector<int>, 2504> G;
int BFS(int s){
    int ans = 1e9;
    queue<pair<int, int>> Q;
    dis[s] = 1;
    Q.push({s, 0});
    while(!Q.empty()){
        auto [u, pre] = Q.front();
        Q.pop();
        for(int v : G[u]){
            if(v == pre) continue;
            if(dis[v]){
                ans = min(ans, dis[u] + dis[v] - 1);
            }else{
                dis[v] = dis[u] + 1;
                Q.push({v, u});
            }
        }
    }
    return ans;
}
signed main(){
    int n, m, a, b, ans = 1e9;
    cin >> n >> m;
    while(m--){
        cin >> a >> b;
        G[a].pb(b);
        G[b].pb(a);
    }
    for(int i = 1; i <= n; i++){
        for(int &d : dis) d = 0;
        ans = min(ans, BFS(i));
    }
    if(ans >= 1e9) cout << "-1\n";
    else cout << ans << "\n";
    return 0;
}
```

Intersection Points (https://cses.fi/problemset/task/1740)

BIT Sweeping Line

```cpp
#include <bits/stdc++.h>
#define int long long
#define pb push_back
using namespace std;
struct line{
    int p, l, r;
};
const int inf = 1e6 + 1;
array<int, 2000004> BIT;
vector<line> A, Q;
bool cmp(line a, line b){
    return a.p < b.p;
}
void update(int p, int x){
    for(; p < 2000004; p += p & -p) BIT[p] += x;
}
int query(int p){
    int sum = 0;
    for(; p; p -= p & -p) sum += BIT[p];
    return sum;
}
int run(){
    int ans = 0, p = 0;
    for(auto [t, l, r] : Q){
        while(p < A.size()){
            auto [x, y, v] = A[p];
            if(x > t) break;
            update(y, v);
            p++;
        }
        ans += query(r) - query(l - 1);
    }
    return ans;
}
signed main(){
    int n, x1, x2, y1, y2;
    cin >> n;
    for(int i = 0; i < n; i++){
        cin >> x1 >> y1 >> x2 >> y2;
        x1 += inf, x2 += inf, y1 += inf, y2 += inf;
        if(x1 == x2) Q.pb({x1, y1, y2});
        else A.pb({x1, y1, 1}), A.pb({x2 + 1, y2, -1});
    }
    sort(Q.begin(), Q.end(), cmp);
    sort(A.begin(), A.end(), cmp);
    cout << run() << "\n";
    return 0;
}
```

Inverse Inversions (https://cses.fi/problemset/task/2214)

Greedy

```cpp
#include <bits/stdc++.h>
#define int long long
using namespace std;
array<int, 1000004> P;
void run(int n, int k){
    int l = 1, r = n;
    for(int i = 1; i <= n; i++){
        if(k >= n - i){
            P[i] = r--;
            k -= n - i;
        }
        else P[i] = l++;
    }
}
signed main(){
    int n, k;
    cin >> n >> k;
    run(n, k);
    for(int i = 1; i <= n; i++) cout << P[i] << " ";
    cout << "\n";
    return 0;
}
```

Monotone Subsequences (https://cses.fi/problemset/task/2215/)

分塊

```cpp
#include <bits/stdc++.h>
using namespace std;

signed main(){
    int t, n, k, l;
    cin >> t;
    while(t--){
        l = 0;
        cin >> n >> k;
        if(k * k < n){
            cout << "IMPOSSIBLE\n";
            continue;
        }
        for(int i = k; i < n + k; i += k){
            for(int j = min(i, n); j > l; j--){
                cout << j << " ";
            }
            l = i;
        }
        cout << "\n";
    }
    return 0;
}
```

String Reorder (https://cses.fi/problemset/task/1743)

Greedy

```cpp
#include <bits/stdc++.h>
using namespace std;
array<int, 128> C;
set<char> A;
string run(string &S){
    int cnt = 0, n = S.size();
    char pre = '#', mos, now;
    string ans;
    for(char c = 'A'; c <= 'Z'; c++){
        if(C[c] > cnt){
            cnt = C[c];
            mos = c;
        }
    }
    while(2 * cnt <= n){
        cnt = 0;
        if(pre == *A.begin()) now = *++A.begin();
        else now = *A.begin();
        C[now]--;
        ans += now;
        if(!C[now]) A.erase(now);
        for(char c = 'A'; c <= 'Z'; c++){
            if(C[c] > cnt){
                cnt = C[c];
                mos = c;
            }
        }
        pre = now;
        n--;
    }
    while(C[mos] > 1){
        ans += mos;
        if(mos == *A.begin()) now = *++A.begin();
        else now = *A.begin();
        ans += now;
        C[now]--;
        C[mos]--;
        if(!C[now]) A.erase(now);
    }
    ans += mos;
    return ans;
}
signed main(){
    int n, cnt = 0;
    string S;
    cin >> S;
    n = S.size();
    for(char s : S){
        C[s]++;
        cnt = max(cnt, C[s]);
        A.insert(s);
    }
    if(cnt * 2 > n + (n & 1)){
```

```
54            cout << "-1\n";
55            return 0;
56        }
57        cout << run(S) << "\n";
58        return 0;
59    }
```

Stack Weights (https://cses.fi/problemset/task/2425)

Segment Tree

```cpp
#include <bits/stdc++.h>
#define mid ((l + r) >> 1)
#define lc (p << 1)
#define rc ((p << 1) | 1)
using namespace std;
array<int, 800004> mix, man, tag;
void pull(int p){
    mix[p] = max(mix[lc], mix[rc]);
    man[p] = min(man[lc], man[rc]);
}
void push(int p){
    mix[lc] += tag[p];
    mix[rc] += tag[p];
    man[lc] += tag[p];
    man[rc] += tag[p];
    tag[lc] += tag[p];
    tag[rc] += tag[p];
    tag[p] = 0;
}
void update(int p, int l, int r, int ql, int qr, int x){
    if(ql > r || qr < l) return;
    if(ql <= l && qr >= r){
        mix[p] += x;
        man[p] += x;
        tag[p] += x;
        return;
    }
    push(p);
    update(lc, l, mid, ql, qr, x);
    update(rc, mid + 1, r, ql, qr, x);
    pull(p);
}
signed main(){
    int n, c, s;
    cin >> n;
    for(int i = 0; i < n; i++){
        cin >> c >> s;
        update(1, 1, n, 1, c, 3 - 2 * s);
        if(mix[1] >= 0 && man[1] >= 0) cout << ">\n";
        else if(mix[1] <= 0 && man[1] <= 0) cout << "<\n";
        else cout << "?\n";
    }
    return 0;
}
```

Pyramid Array (https://cses.fi/problemset/task/1747)

BIT

```cpp
#include <bits/stdc++.h>
#define int long long
#define pb push_back
using namespace std;
array<int, 200004> BIT;
void update(int p, int x){
    for(; p < 200004; p += p & -p) BIT[p] += x;
}
int query(int p){
    int sum = 0;
    for(; p; p -= p & -p) sum += BIT[p];
    return sum;
}
signed main(){
    int n, x, ans = 0;
    vector<pair<int, int>> S;
    cin >> n;
    for(int i = 1; i <= n; i++){
        cin >> x;
        S.pb({x, i});
        update(i, 1);
    }
    sort(S.begin(), S.end());
    for(auto [v, p] : S){
        ans += min(query(p - 1), query(200000) - query(p));
        update(p, -1);
    }
    cout << ans << "\n";
    return 0;
}
```

Increasing Subsequence II (https://cses.fi/problemset/task/1748)

BIT DP

```cpp
#include <bits/stdc++.h>
#define int long long
#define pb push_back
using namespace std;
const int mod = 1e9 + 7;
array<int, 200004> X, BIT;
void update(int p, int x){
    for(; p < 200004; p += p & -p) BIT[p] += x;
}
int query(int p){
    int sum = 0;
    for(; p; p -= p & -p) sum = (sum + BIT[p]) % mod;
    return sum;
}
void umbrella(vector<pair<int, int>> &S){
    int lst = 0, cnt = 1;
    sort(S.begin(), S.end());
    for(auto [x, p] : S){
        if(x == lst) X[p] = cnt;
        else X[p] = ++cnt;
        lst = x;
    }
}
int DP(int n){
    int ans = 0, cnt;
    update(1, 1);
    for(int i = 1; i <= n; i++){
        cnt = query(X[i] - 1);
        update(X[i], cnt);
        ans = (ans + cnt) % mod;
    }
    return ans;
}
signed main(){
    int n, x;
    vector<pair<int, int>> S;
    cin >> n;
    for(int i = 1; i <= n; i++){
        cin >> x;
        S.pb({x, i});
    }
    umbrella(S);
    cout << DP(n) << "\n";
    return 0;
}
```

String Removals (https://cses.fi/problemset/task/1149)

DP

```cpp
#include <bits/stdc++.h>
#define int long long
using namespace std;
int mod = 1e9 + 7;
array<int, 500004> dp;
int DP(string &S){
    int n = S.size(), ans = 0;
    dp[0] = 1;
    for(int i = 1; i <= n; i++){
        for(int j = i - 1; j >= 0; j--){
            dp[i] = (dp[i] + dp[j]) % mod;
            if(!j || S[i - 1] == S[j - 1]) break;
        }
        ans = (ans + dp[i]) % mod;
    }
    return ans;
}
signed main(){
    string S;
    cin >> S;
    cout << DP(S) << "\n";
    return 0;
}
```

Bit Inversions (https://cses.fi/problemset/task/1188)

Segment Tree

```cpp
#include <bits/stdc++.h>
#define mid ((l + r) >> 1)
#define lc (p << 1)
#define rc ((p << 1) | 1)
using namespace std;
array<array<int, 800004>, 2> seg, pre, suf, all;
void pull(int p, int l, int r){
    for(int i : {0, 1}){
        all[i][p] = all[i][lc] & all[i][rc];
        if(all[i][p]){
            pre[i][p] = suf[i][p] = seg[i][p] = r - l + 1;
            continue;
        }
        if(all[i][lc]) pre[i][p] = (mid - l + 1) + pre[i][rc];
        else pre[i][p] = pre[i][lc];
        if(all[i][rc]) suf[i][p] = (r - mid) + suf[i][lc];
        else suf[i][p] = suf[i][rc];
        seg[i][p] = max({seg[i][lc], seg[i][rc], pre[i][p], suf[i][
    }
}
void update(int p, int l, int r, int c, int x){
    if(c > r || c < l) return;
    if(l == r){
        seg[x][p] = pre[x][p] = suf[x][p] = all[x][p] = 1;
        seg[x ^ 1][p] = pre[x ^ 1][p] = suf[x ^ 1][p] = all[x ^ 1][
        return;
    }
    update(lc, l, mid, c, x);
    update(rc, mid + 1, r, c, x);
    pull(p, l, r);
}
signed main(){
    int n, q, p;
    string S;
    cin >> S;
    n = S.size();
    for(int i = 1; i <= n; i++){
        update(1, 1, n, i, S[i - 1] ^ 48);
    }
    cin >> q;
    while(q--){
        cin >> p;
        update(1, 1, n, p, S[p - 1] ^ 49);
        cout << max(seg[0][1], seg[1][1]) << " ";
        S[p - 1] ^= 1;
    }
    cout << "\n";
    return 0;
}
```

Xor Pyramid (https://cses.fi/problemset/task/2419)

Lucas Theorem

```cpp
#include <bits/stdc++.h>
using namespace std;
int C(int n, int k){
    if(n == k || !k) return 1;
    if(k > n) return 0;
    return C(n >> 1, k >> 1) & C(n & 1, k & 1);
}
signed main(){
    int n, x, sum = 0;
    cin >> n;
    for(int i = 0; i < n; i++){
        cin >> x;
        if(C(n - 1, i)) sum ^= x;
    }
    cout << sum << "\n";
    return 0;
}
```

Writing Numbers (https://cses.fi/problemset/task/1086)

二分搜

```cpp
#include <bits/stdc++.h>
#define int long long
using namespace std;
int see(int n){
    int ans = 0, cnt;
    for(int i = 1; i < 10; i++){
        cnt = 0;
        for(int t = 1; t <= n; t *= 10){
            cnt += (n / (10 * t)) * t;
            cnt += min(t, max(0ll, n % (10 * t) - i * t + 1));
        }
        ans = max(ans, cnt);
    }
    return ans;
}
int BIS(int k){
    int l = 0, r = 1e18, mid;
    while(l != r){
        mid = ((l + r) >> 1) + 1;
        if(see(mid) <= k) l = mid;
        else r = mid - 1;
    }
    return l;
}
signed main(){
    int n;
    cin >> n;
    cout << BIS(n) << "\n";
    return 0;
}
```

String Transform (https://cses.fi/problemset/task/1113)

爆搜

```cpp
#include <bits/stdc++.h>
#define pb push_back
using namespace std;
array<vector<char>, 128> C;
array<vector<int>, 128> I;
array<int, 128> P;
string trans(string  &S){
    int p = 0;
    char now = '#', tmp;
    string ans;
    for(char s : S){
        C[s].pb(s);
        I[s].pb(0);
    }
    for(int i = 0; i < 128; i++){
        for(int j = 0; j < C[i].size(); j++){
            C[i][j] = S[p];
            I[i][j] = P[S[p++]]++;
        }
    }
    p = 0;
    for(int i = 1; i < S.size(); i++){
        tmp = now;
        now = C[now][p];
        p = I[tmp][p];
        ans += now;
    }
    reverse(ans.begin(), ans.end());
    return ans;
}
signed main(){
    string S;
    cin >> S;
    cout << trans(S) << "\n";
    return 0;
}
```

Letter Pair Move Game (https://cses.fi/problemset/task/2427)

```
嗷嗷待哺
```

Maximum Building I (https://cses.fi/problemset/task/1147)

Monotone Stack

```cpp
#include <bits/stdc++.h>
#define ff first
#define ss second
using namespace std;
array<int, 1004> H;
array<array<char, 1004>, 1004> G;
stack<pair<int, int>> S;
int run(int n, int m){
    int ans = 0, now;
    for(int j = 1; j <= n; j++){
        S.push({0, 0});
        for(int i = 1; i <= m; i++){
            now = i;
            if(G[j][i] == '*') H[i] = 0;
            else H[i]++;
            while(!S.empty() && H[i] <= S.top().ff){
                auto [h, p] = S.top();
                S.pop();
                ans = max(ans, (i - p) * h);
                now = p;
            }
            S.push({H[i], now});
        }
        while(!S.empty()){
            auto [h, p] = S.top();
            S.pop();
            ans = max(ans, (m - p + 1) * h);
        }
    }
    return ans;
}
signed main(){
    int n, m;
    cin >> n >> m;
    for(int i = 1; i <= n; i++){
        for(int j = 1; j <= m; j++){
            cin >> G[i][j];
        }
    }
    cout << run(n, m) << "\n";
    return 0;
}
```

Sorting Methods (https://cses.fi/problemset/task/1162/)

BIT LIS DFS

```cpp
#include <bits/stdc++.h>
#define int long long
using namespace std;
bitset<200004> vis;
array<int, 200004> P, pre, mix;
void update(int t, int p, int x){
    for(; p < 200004; p += p & -p){
        if(t) mix[p] = max(mix[p], x);
        else pre[p] += x;
    }
}
int query(int t, int p){
    int ans = 0;
    for(; p; p -= p & -p){
        if(t) ans = max(ans, mix[p]);
        else ans += pre[p];
    }
    return ans;
}
void DFS(int u){
    if(vis[u]) return;
    vis[u] = 1;
    DFS(P[u]);
}
signed main(){
    int n, lis = 0, inv = 0, cyc = 0, dec = 0;
    cin >> n;
    for(int i = 1; i <= n; i++){
        cin >> P[i];
        update(0, P[i], 1);
        inv += query(0, 200000) - query(0, P[i]);
        lis = max(lis, query(1, P[i]) + 1);
        update(1, P[i], query(1, P[i]) + 1);
    }
    for(int i = n, p = n; i; i--){
        if(!vis[i]){
            DFS(i);
            cyc++;
        }
        if(P[i] == p){
            p--;
            dec++;
        }
    }
    cout << inv << " " << n - cyc << " " << n - lis << " " << n - d
    return 0;
}
```

Cyclic Array (https://cses.fi/problemset/task/1191)

三分搜

```cpp
#include <bits/stdc++.h>
#define int long long
using namespace std;
int n, k;
array<int, 200004> X;
int see(int p){
    int sum = 0, cnt = 0, s, t;
    for(s = 1; s <= n; s++){
        if(sum + X[s] > p) break;
        sum += X[s];
    }
    for(t = n; t >= s; t--){
        if(sum + X[t] > k) break;
        sum += X[t];
    }
    if(sum) cnt++;
    sum = 0;
    for(int i = s; i <= t; i++){
        if(sum + X[i] > k){
            sum = 0;
            cnt++;
        }
        sum += X[i];
    }
    if(sum) cnt++;
    return cnt;
}
int TIS(){
    int l = 0, r = k, lm, rm, ans = 1e9;
    while(r - l > 2){
        lm = (2 * l + r) / 3ll;
        rm = (l + 2 * r) / 3ll;
        if(see(lm) < see(rm)) r = rm;
        else l = lm;
    }
    for(int i = l; i <= r; i++){
        ans = min(ans, see(i));
    }
    return ans;
}
signed main(){
    cin >> n >> k;
    for(int i = 1; i <= n; i++){
        cin >> X[i];
    }
    cout << TIS() << "\n";
    return 0;
}
```

List of Sums (https://cses.fi/problemset/task/2414/)

爆搜

```cpp
#include <bits/stdc++.h>
#define pb push_back
using namespace std;
array<int, 104> A;
vector<int> B;
bool gen(int n, int a1a2){
    int a0ai;
    multiset<int> BB;
    for(int b : B) BB.insert(b);
    A[0] = (B[0] + B[1] - a1a2) / 2;
    A[1] = B[0] - A[0];
    A[2] = B[1] - A[0];
    BB.erase(BB.find(B[0]));
    BB.erase(BB.find(B[1]));
    BB.erase(BB.find(a1a2));
    for(int i = 3; i < n; i++){
        a0ai = *BB.begin();
        A[i] = a0ai - A[0];
        for(int j = 0; j < i; j++){
            if(BB.find(A[i] + A[j]) == BB.end()) return 0;
            BB.erase(BB.find(A[i] + A[j]));
        }
    }
    for(int i = 1; i < n; i++){
        if(A[i] < A[i - 1]) return 0;
    }
    return 1;
}
signed main(){
    int n, b;
    cin >> n;
    for(int i = 0; i < n * (n - 1) / 2; i++){
        cin >> b;
        B.pb(b);
    }
    sort(B.begin(), B.end());
    for(int i = 2; i <= n; i++){
        if(gen(n, B[i])){
            for(int j = 0; j < n; j++) cout << A[j] << " ";
            break;
        }
    }
    cout << "\n";
    return 0;
}
```

Increasing Array II (https://cses.fi/problemset/task/2132/)

DP圖

```cpp
#include <bits/stdc++.h>
#define int long long
using namespace std;

signed main(){
    int n, x, ans = 0;
    priority_queue<int> Q;
    cin >> n;
    for(int i = 0; i < n; i++){
        cin >> x;
        Q.push(x);
        ans += Q.top() - x;
        Q.pop();
        Q.push(x);
    }
    cout << ans << "\n";
    return 0;
}
```

Food Division (https://cses.fi/problemset/task/1189/)

三分搜

```cpp
#include <bits/stdc++.h>
#define int long long
using namespace std;
array<int, 200004> A, B, cur;
int run(int n, int p){
    int cnt = llabs(p);
    for(int i = 1; i <= n; i++){
        cur[i] = A[i];
    }
    cur[n] -= p;
    cur[1] += p;
    for(int i = 1; i < n; i++){
        cnt += llabs(cur[i] - B[i]);
        cur[i + 1] += cur[i] - B[i];
        cur[i] = B[i];
    }
    return cnt;
}
int TIS(int n){
    int l = -1e12, r = 1e12, lmid, rmid;
    while(r - l > 2){
        lmid = (2 * l + r) / 3;
        rmid = (l + 2 * r) / 3;
        if(run(n, lmid) < run(n, rmid)) r = rmid;
        else l = lmid;
    }
    return min({run(n, l), run(n, l + 1), run(n, r)});
}
signed main(){
    int n;
    cin >> n;
    for(int i = 1; i <= n; i++) cin >> A[i];
    for(int i = 1; i <= n; i++) cin >> B[i];
    cout << TIS(n) << "\n";
    return 0;
}
```

Bit Problem (https://cses.fi/problemset/task/1654/)

SOS DP

```cpp
#include <bits/stdc++.h>
using namespace std;
const int C = (1 << 20) - 1;
array<int, 200004> X, O, A, R;
array<int, 1 << 20> dp;
void OR(int n){
    for(int &d : dp) d = 0;
    for(int i = 1; i <= n; i++) dp[X[i]]++;
    for(int k = 1; k < 1 << 20; k <<= 1){
        for(int i = 0; i < 1 << 20; i++){
            if(i & k) dp[i] += dp[i ^ k];
        }
    }
    for(int i = 1; i <= n; i++){
        O[i] = dp[X[i]];
        R[i] = n - dp[C ^ X[i]];
    }
}
void AND(int n){
    for(int &d : dp) d = 0;
    for(int i = 1; i <= n; i++) dp[X[i]]++;
    for(int k = 1 << 19; k; k >>= 1){
        for(int i = 0; i < 1 << 20; i++){
            if(i & k) continue;
            dp[i] += dp[i ^ k];
        }
    }
    for(int i = 1; i <= n; i++){
        A[i] = dp[X[i]];
    }
}
signed main(){
    int n;
    cin >> n;
    for(int i = 1; i <= n; i++) cin >> X[i];
    OR(n);
    AND(n);
    for(int i = 1; i <= n; i++){
        cout << O[i] << " " << A[i] <<" " << R[i] << "\n";
    }
    return 0;
}
```

Swap Round Sorting (https://cses.fi/problemset/task/1698)

通靈

```cpp
#include <bits/stdc++.h>
#define pb push_back
using namespace std;
int cnt = 0;
bitset<200004> vis;
array<int, 200004> G, R;
array<vector<pair<int, int>>, 200004> ans;
void cut(int u){
    int l, r, nl, nr;
    if(G[u] == u || vis[u]) return;
    if(R[u] == G[u]){
        if(vis[G[u]]) return;
        ans[cnt].pb({u, G[u]});
        vis[u] = vis[G[u]] = 1;
        swap(G[u], G[G[u]]);
        swap(R[u], R[R[u]]);
    }else{
        while(1){
            l = R[u], r = G[u];
            nl = R[l], nr = G[r];
            ans[cnt].pb({l, r});
            vis[l] = vis[r] = 1;
            if(nr == l && nl == r){
                R[l] = l;
                R[u] = r;
                swap(G[l], G[r]);
                return;
            }else{
                R[u] = r;
                R[nr] = l;
                swap(G[l], G[r]);
                u = l;
            }
            if(nl == nr) return;
            u = l;
        }
    }
}
void run(int n){
    bool ok;
    while(1){
        ok = 1;
        for(int i = 1; i <= n; i++){
            vis[i] = 0;
            ok &= G[i] == i;
        }
        if(ok) return;
        for(int i = 1; i <= n; i++) cut(i);
        cnt++;
    }
}
signed main(){
    int n, x;
```

```
    cin >> n;
    for(int i = 1; i <= n; i++){
        cin >> x;
        G[i] = x;
        R[x] = i;
    }
    run(n);
    cout << cnt << "\n";
    for(int i = 0; i < cnt; i++){
        cout << ans[i].size() << "\n";
        for(auto [u, v] : ans[i]){
            cout << u << " " << v << "\n";
        }
    }
    return 0;
}
```

Binary Subsequences (https://cses.fi/problemset/task/2430)

逆DP

```cpp
#include <bits/stdc++.h>
using namespace std;
const int inf = 1 << 20;
string ans;
int RUN(int a, int b){
    if(!a && !b) return 0;
    if(a == b) return inf;
    return RUN(a % (b + 1), b % (a + 1)) + a / (b + 1) + b / (a + 1
}
void back(int a, int b){
    if(!a && !b) return;
    if(a > b){
        back(a - b - 1, b);
        ans += '0';
    }
    else{
        back(a, b - a - 1);
        ans += '1';
    }
}
signed main(){
    int n, cnt = inf, tmp, a, b;
    cin >> n;
    for(int i = 0; i <= n; i++){
        tmp = RUN(i, n - i);
        if(tmp < cnt){
            a = i, b = n - i;
            cnt = tmp;
        }
    }
    back(a, b);
    cout << ans << "\n";
    return 0;
}
```

Tree Isomorphism I (https://cses.fi/problemset/task/1700)

Hash

```cpp
#include <bits/stdc++.h>
#define int long long
#define pb push_back
using namespace std;
const int mod = 1000696969, c = 41;
array<int, 100004> C, S1, S2;
array<vector<int>, 100004> T1, T2;
void build(int n){
    C[0] = 1;
    for(int i = 1; i <= n; i++){
        C[i] = c * C[i - 1] % mod;
    }
}
int DFS(array<vector<int>, 100004> &T, array<int, 100004> &S, int u
    int sum = 0;
    vector<pair<int, int>> H;
    for(int v : T[u]){
        if(v == pre) continue;
        H.pb({DFS(T, S, v, u), S[v]});
    }
    sort(H.begin(), H.end());
    for(auto [h, s] : H){
        sum = (sum + h * C[S[u]]) % mod;
        S[u] += s;
    }
    S[u]++;
    sum = (sum + S[u] * C[S[u]]) % mod;
    return sum;
}
signed main(){
    int t, n, a, b;
    build(100000);
    cin >> t;
    while(t--){
        cin >> n;
        for(int i = 1; i <= n; i++){
            S1[i] = S2[i] = 0;
            T1[i].clear();
            T2[i].clear();
        }
        for(int i = 1; i < n; i++){
            cin >> a >> b;
            T1[a].pb(b);
            T1[b].pb(a);
        }
        for(int i = 1; i < n; i++){
            cin >> a >> b;
            T2[a].pb(b);
            T2[b].pb(a);
        }
        if(DFS(T1, S1, 1, 0) == DFS(T2, S2, 1, 0)) cout << "YES\n";
        else cout << "NO\n";
    }
```

```
    return 0;
}
```

Counting Sequences (https://cses.fi/problemset/task/2228/)

Math

```
#include <bits/stdc++.h>
#define int long long
using namespace std;
const int mod = 1e9 + 7;
array<int, 1000004> fac;
void build(int n){
    fac[0] = 1;
    for(int i = 1; i <= n; i++){
        fac[i] = i * fac[i - 1] % mod;
    }
}
int exp(int x, int k){
    int ans = 1;
    for(int i = 1; i <= k; i <<= 1){
        if(i & k) ans = ans * x % mod;
        x = x * x % mod;
    }
    return ans;
}
int C(int n, int k){
    return (fac[n] * exp(fac[n - k], mod - 2) % mod) * exp(fac[k],
}
int math(int n, int k){
    int ans = 0;
    for(int i = 0; i <= k; i++){
        ans = (ans + ((exp(-1, k - i) * C(k, i) % mod) * exp(i, n)
    }
    return ans;
}
signed main(){
    int n, k;
    cin >> n >> k;
    build(1000000);
    cout << math(n, k) << "\n";
    return 0;
}
```

Critical Cities (https://cses.fi/problemset/task/1703)

Dominator Tree

```cpp
#include <bits/stdc++.h>
#define pb push_back
using namespace std;
int cnt = 0;
array<int, 100004> P, rev, L, DSU, pre, sdom, idom;
array<vector<int>, 100004> G, R, buk;
vector<int> ans;
int find(int u, int x = 0){
    if(DSU[u] == u) return x? -1 : u;
    int v = find(DSU[u], x + 1);
    if(v < 0) return u;
    if(sdom[L[DSU[u]]] < sdom[L[u]]) L[u] = L[DSU[u]];
    DSU[u] = v;
    return x? v : L[u];
}
void onion(int u, int v){
    DSU[v] = u;
}
void DFS(int u){
    P[u] = ++cnt, rev[cnt] = u, L[cnt] = cnt, DSU[cnt] = cnt, sdom[
    for(int v : G[u]){
        if(!P[v]){
            DFS(v);
            pre[P[v]] = P[u];
        }
        R[P[v]].pb(P[u]);
    }
}
void run(int u){
    ans.pb(rev[u]);
    if(u == 1) return;
    run(idom[u]);
}
signed main(){
    int n, m, a, b, w;
    cin >> n >> m;
    while(m--){
        cin >> a >> b;
        G[a].pb(b);
    }
    DFS(1);
    for(int u = n; u; u--){
        for(int v : R[u]){
            sdom[u] = min(sdom[u], sdom[find(v)]);
        }
        if(u > 1) buk[sdom[u]].pb(u);
        for(int v : buk[u]){
            w = find(v);
            if(sdom[v] == sdom[w]) idom[v] = sdom[v];
            else idom[v] = w;
        }
        onion(pre[u], u);
    }
```

```
54      for(int u = 2; u <= n; u++){
55          if(idom[u] != sdom[u]) idom[u] = idom[idom[u]];
56      }
57      run(P[n]);
58      sort(ans.begin(), ans.end());
59      cout << ans.size() << "\n";
60      for(int u : ans) cout << u << " ";
61      cout << "\n";
62      return 0;
63  }
```

School Excursion (https://cses.fi/problemset/task/1706)

DP DSU

```cpp
#include <bits/stdc++.h>
using namespace std;
int p = 0;
array<int, 100004> DSU, S, C;
array<array<int, 100004>, 2> dp;
int find(int u){
    if(DSU[u] == u) 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;
    if(S[U] < S[V]) swap(U, V);
    DSU[V] = U;
    S[U] += S[V];
}
void DP(int n){
    int sum;
    dp[0][0] = 1;
    for(int i = 1; i <= n; i++){
        if(!C[i]) continue;
        p ^= 1;
        for(int j = 0; j < i; j++){
            sum = 0;
            for(int k = j; k <= n; k += i){
                if(k - j > i * C[i]) sum -= dp[p ^ 1][k - i * (C[i]
                sum += dp[p ^ 1][k];
                dp[p][k] = sum? 1 : 0;
            }
        }
    }
}
signed main(){
    int n, m, a, b;
    cin >> n >> m;
    for(int i = 1; i <= n; i++){
        DSU[i] = i;
        S[i] = 1;
    }
    while(m--){
        cin >> a >> b;
        onion(a, b);
    }
    for(int i = 1; i <= n; i++){
        if(DSU[i] != i) continue;
        C[S[i]]++;
    }
    DP(n);
    for(int i = 1; i <= n; i++) cout << dp[p][i];
    cout << "\n";
    return 0;
}
```

Coin Grid (https://cses.fi/problemset/task/1709/)

Dinic

```cpp
#include <bits/stdc++.h>
#define pb push_back
using namespace std;
struct edge{
    int u, v, f;
};
int inf = 1000696969, cnt = 0;
bitset<204> vis;
array<int, 204> lvl, P;
array<edge, 84004> E;
array<vector<int>, 204> G;
void add(int u, int v, int f){
    E[cnt] = {u, v, 1};
    G[u].pb(cnt++);
    E[cnt] = {v, u, 0};
    G[v].pb(cnt++);
}
bool BFS(int s, int t){
    int u;
    queue<int> Q;
    Q.push(s);
    lvl[s] = 1;
    while(!Q.empty()){
        u = Q.front();
        Q.pop();
        for(int i : G[u]){
            auto [o, v, f] = E[i];
            if(!f || lvl[v]) continue;
            lvl[v] = lvl[u] + 1;
            Q.push(v);
        }
    }
    return lvl[t];
}
int DFS(int u, int t, int f){
    if(u == t || !f) return f;
    int wat = 0, tmp;
    for(int &i = P[u]; i < G[u].size(); i++){
        auto &[eu, ev, ef] = E[G[u][i]];
        auto &[bu, bv, bf] = E[G[u][i] ^ 1];
        if(lvl[ev] != lvl[u] + 1) continue;
        tmp = DFS(ev, t, min(f, ef));
        ef -= tmp, bf += tmp;
        f -= tmp, wat += tmp;
    }
    return wat;
}
int flow(int s, int t){
    int tmp, ans = 0;
    while(1){
        for(int &l : lvl) l = 0;
        if(!BFS(s, t)) break;
        while(1){
```

```
            for(int &p : P) p = 0;
            if(tmp = DFS(s, t, inf)) ans += tmp;
            else break;
        }
    }
    return ans;
}
void run(int u){
    if(vis[u]) return;
    vis[u] = 1;
    for(int i : G[u]){
        auto [o, v, f] = E[i];
        if(!f) continue;
        run(v);
    }
}
signed main(){
    int n;
    char c;
    cin >> n;
    for(int i = 1; i <= n; i++){
        add(0, i, 1);
        add(100 + i, 201, 1);
    }
    for(int i = 1; i <= n; i++){
        for(int j = 1; j <= n; j++){
            cin >> c;
            if(c == '.') continue;
            add(i, 100 + j, 1);
        }
    }
    cout << flow(0, 201) << "\n";
    run(0);
    for(int i = 1; i <= n; i++){
        if(!vis[i]) cout << "1 " << i << "\n";
        if(vis[100 + i]) cout << "2 " << i << "\n";
    }
    return 0;
}
```

Robot Path (https://cses.fi/problemset/task/1742)

Binary Search Segment Tree

```
#include <bits/stdc++.h>
#define int long long
#define pb push_back
using namespace std;
struct line{
    int t, p, l, r;
};
int N = 1 << 18;
array<char, 100004> F;
array<int, 1 << 19> seg, tag, D;
array<vector<pair<int, int>>, 200004> X, Y;
vector<int> P;
vector<line> S, Q;
map<int, int> O;
bool cmp(line a, line b){
    if(a.p == b.p) return abs(a.t) > abs(b.t);
    return a.p < b.p;
}
void update(int l, int r, int x){
    int cntl = 0, cntr = 0;
    l += N - 1, r += N + 1;
    for(int n = 1; (l ^ r) > 1; l >>= 1, r >>= 1, n <<= 1){
        seg[l] += cntl * x, seg[r] += cntr * x;
        if(~l & 1) tag[l ^ 1] += x, seg[l ^ 1] += n * x, cntl += n
        if(r & 1) tag[r ^ 1] += x, seg[r ^ 1] += n * x, cntr += n;
    }
    for(; l; l >>= 1, r >>= 1) seg[l] += cntl * x, seg[r] += cntr
}
int query(int l, int r){
    int sum = 0, cntl = 0, cntr = 0;
    l += N - 1, r += N + 1;
    for(int n = 1; (l ^ r) > 1; l >>= 1, r >>= 1, n <<= 1){
        sum += tag[l] * cntl + tag[r] * cntr;
        if(~l & 1) sum += seg[l ^ 1], cntl += n;
        if(r & 1) sum += seg[r ^ 1], cntr += n;
    }
    for(; r; r >>= 1) sum += cntl * tag[l] + cntr * tag[r];
    return sum;
}
bool banana(int p){
    int ans = 0;
    Q.clear();
    for(int i = 0; i < p; i++){
        auto [x1, x2, y1, y2] = S[i];
        if(x1 == x2){
            if(y1 > y2) swap(y1, y2);
            Q.pb({0, x1, y1, y2});
            X[x2].pb({y1, y2});
        }else{
            if(x1 > x2) swap(x1, x2);
            Q.pb({1, x1, y1, 0});
            Q.pb({-1, x2 + 1, y2, 0});
            Y[y1].pb({x1, x2});
```

```
        }
    }
    sort(Q.begin(), Q.end(), cmp);
    for(auto [t, p, l, r] : Q){
        if(t) update(l, l, t);
        else ans |= query(l, r);
    }
    for(int i = 1; i < 200004; i++){
        for(auto [l, r] : X[i]){
            ans |= query(l, r);
            update(l, r, 1);
        }
        for(auto [l, r] : X[i]){
            update(l, r, -1);
        }
        for(auto [l, r] : Y[i]){
            ans |= query(l, r);
            update(l, r, 1);
        }
        for(auto [l, r] : Y[i]){
            update(l, r, -1);
        }
        X[i].clear(), Y[i].clear();
    }
    return ans;
}
int crash(int n){
    if(F[n] == 'U' && F[n - 1] == 'D' || F[n] == 'D' && F[n - 1] =
    if(F[n] == 'L' && F[n - 1] == 'R' || F[n] == 'R' && F[n - 1] =
    auto [sx, tx, sy, ty] = S[n];
    for(int i = 0; i < n; i++){
        auto [x1, x2, y1, y2] = S[i];
        if(y1 > y2) swap(y1, y2);
        if(x1 > x2) swap(x1, x2);
        if(sy == ty){
            if(y1 <= sy && y2 >= ty) update(x1, x2, 1);
        }else{
            if(x1 <= sx && x2 >= tx) update(y1, y2, 1);
        }
    }
    if((sy == ty && sx <= tx) || (sx == tx && sy <= ty)){
        for(int i = (sx == tx? sy : sx); i <= (sx == tx? ty : tx);
            if(query(i, i)) return O[i] - O[(sx == tx? sy : sx)] +
        }
    }else{
        for(int i = (sx == tx? sy : sx); i >= (sx == tx? ty : tx);
            if(query(i, i)) return O[(sx == tx? sy : sx)] - O[i] +
        }
    }
    return 1;
}
int BIS(int n){
    int p = 0;
    for(int i = 1 << 16; i; i >>= 1){
```

```
    for(int i = 1 << 16; i; i >>= 1){
        if(p + i <= n && !banana(p + i)) p += i;
    }
    return p;
}
void crepe(vector<int> &X, vector<line> &L){
    int p = 0;
    map<int, int> M;
    for(int x : X) M[x] = 0;
    for(auto &[x, c] : M) c = ++p, O[p] = x;
    for(auto &[x1, x2, y1, y2] : L){
        x1 = M[x1], x2 = M[x2];
        y1 = M[y1], y2 = M[y2];
    }
}
signed main(){
    cin.tie(0), cout.tie(0), ios::sync_with_stdio(0);
    char d;
    int n, t, x = 0, y = 0, px, py, p;
    cin >> n;
    P.pb(0);
    for(int i = 1; i <= n; i++){
        cin >> d >> t;
        D[i] = D[i - 1] + t;
        if(d == 'U' || d == 'D'){
            py = y + (d == 'U'? 1 : -1), y += (d == 'U'? t : -t);
            if(i == 1) py += (d == 'U'? -1 : 1);
            S.pb({x, x, py, y});
            P.pb(py), P.pb(y);
        }else{
            px = x + (d == 'R'? 1 : -1), x += (d == 'R'? t : -t);
            if(i == 1) px += (d == 'R'? -1 : 1);
            S.pb({px, x, y, y});
            P.pb(px), P.pb(x);
        }
        F[i - 1] = d;
    }
    crepe(P, S);
    p = BIS(n);
    if(p == n) cout << D[n] << "\n";
    else cout << D[p] + crash(p) << "\n";
    return 0;
}
```

Programmers and Artists (https://cses.fi/problemset/task/2426)

Greedy

```cpp
#include <bits/stdc++.h>
#define int long long
#define pb push_back
#define pii pair<int, int>
#define ff first
#define ss second
using namespace std;
bitset<200004> pro, art, awy;
array<int, 200004> X, Y;
vector<pair<int, int>> P, A;
priority_queue<int> sas;
priority_queue<int, vector<int>, greater<int>> bas;
priority_queue<pii> de;
int chos(int a, int b){
    int sum = 0, ans = 0;
    for(int i = 0; i < a + b; i++){
        auto [x, j] = P[i];
        sum += Y[j];
        de.push({x - Y[j], j});
        art[j] = 1;
    }
    for(int i = a + b; i < P.size(); i++){
        auto [x, j] = P[i];
        sas.push(Y[j]);
    }
    sas.push(0);
    while(de.size() > b){
        auto [d, i] = de.top();
        de.pop();
        sum += d;
        pro[i] = 1;
        art[i] = 0;
    }
    ans = max(ans, sum);
    for(int i = a + b - 1; i >= a; i--){
        auto [x, j] = P[i];
        awy[j] = 1;
        if(art[j]){
            art[j] = 0;
            sum -= Y[j];
        }else{
            pro[j] = 0;
            sum -= x;
            while(awy[de.top().ss]) de.pop();
            auto [d, k] = de.top();
            de.pop();
            sum += d;
            pro[k] = 1;
            art[k] = 0;
        }
        bas.push(Y[j]);
        sum += Y[j];
        if(sas.top() > bas.top()){
```

```
            sum += sas.top() - bas.top();
            sas.push(bas.top());
            bas.push(sas.top());
            bas.pop(), sas.pop();
        }
        ans = max(ans, sum);
    }
    return ans;
}
signed main(){
    int a, b, n;
    cin >> a >> b >> n;
    for(int i = 1; i <= n; i++){
        cin >> X[i] >> Y[i];
        P.pb({X[i], i});
        A.pb({Y[i], i});
    }
    sort(P.begin(), P.end(), greater<pii>());
    sort(A.begin(), A.end(), greater<pii>());
    cout << chos(a, b) << "\n";
    return 0;
}
```

Course Schedule II (https://cses.fi/problemset/task/1757)

Topological Sort

```cpp
#include <bits/stdc++.h>
#define pb push_back
using namespace std;
array<int, 100004> in;
array<vector<int>, 100004> G;
vector<int> ord;
void topu(int n){
    int u;
    priority_queue<int> Q;
    for(int i = 1; i <= n; i++){
        if(!in[i]) Q.push(i);
    }
    while(!Q.empty()){
        u = Q.top();
        Q.pop();
        ord.pb(u);
        for(int v : G[u]){
            if(!--in[v]) Q.push(v);
        }
    }
}
signed main(){
    int n, m, a, b;
    cin >> n >> m;
    while(m--){
        cin >> a >> b;
        in[a]++;
        G[b].pb(a);
    }
    topu(n);
    reverse(ord.begin(), ord.end());
    for(int u : ord) cout << u << " ";
    cout << "\n";
    return 0;
}
```

Removing Digits II (https://cses.fi/problemset/task/2174)

```
嗷嗷待哺
```

Coin Arrangement (https://cses.fi/problemset/task/2180)

Greedy

```cpp
#include <bits/stdc++.h>
#define int long long
using namespace std;
array<array<int, 3>, 100004> C;
int RUN(int n){
    int cnt = 0;
    for(int i = 1; i <= n; i++){
        if(C[i][1] * C[i][2] < 0){
            if(abs(C[i][1]) > abs(C[i][2])){
                cnt += abs(C[i][2]);
                C[i][1] += C[i][2];
                C[i][2] = 0;
            }else{
                cnt += abs(C[i][1]);
                C[i][2] += C[i][1];
                C[i][1] = 0;
            }
        }
        cnt += abs(C[i][1] + C[i][2]);
        C[i + 1][1] += C[i][1], C[i + 1][2] += C[i][2];
    }
    return cnt;
}
signed main(){
    int n;
    cin >> n;
    for(int i : {1, 2}){
        for(int j = 1; j <= n; j++){
            cin >> C[j][i];
            C[j][i] -= 1;
        }
    }
    cout << RUN(n) << "\n";
    return 0;
}
```

Counting Bishops (https://cses.fi/problemset/task/2176)

DP

```cpp
#include <bits/stdc++.h>
#define int long long
using namespace std;
const int mod = 1e9 + 7, two = 5e8 + 4;
array<array<int, 250004>, 504> dp;
int DP(int n, int k){
    int ans = 0;
    dp[1][0] = 1, dp[1][1] = 2;
    dp[2][0] = 1, dp[2][1] = 4, dp[2][2] = 2;
    for(int i = 3; i < n; i++){
        for(int j = 0; j <= min(i, k); j++){
            dp[i][j] = (dp[i][j] + dp[i - 2][j]) % mod;
            if(j) dp[i][j] = (dp[i][j] + dp[i - 2][j - 1] * 2 * (i
            if(j >= 2) dp[i][j] = (dp[i][j] + (dp[i - 2][j - 2] * 2
        }
    }
    if(n > 2){
        for(int j = 0; j <= min(n, k); j++){
            dp[n][j] = (dp[n][j] + dp[n - 2][j]) % mod;
            dp[n][j] = (dp[n][j] + dp[n - 2][j - 1] * (n - j + 1))
        }
    }
    if(n == 1) return 1;
    if(n == 2){
        if(k == 0) return 1;
        else if(k == 1 || k == 2) return 4;
        else return 0;
    }
    for(int i = 0; i <= k; i++){
        ans = (ans + dp[n][i] * dp[n - 1][k - i]) % mod;
    }
    return ans;
}
signed main(){
    int n, k;
    cin >> n >> k;
    cout << DP(n, k) << "\n";
    return 0;
}
```

Grid Puzzle I (https://cses.fi/problemset/task/2432)

Flow

```cpp
#include <bits/stdc++.h>
#define pb push_back
using namespace std;
struct edge{
    int u, v, f;
};
const int inf = 1000696969;
int cnt = 0;
array<int, 104> lvl, P;
array<edge, 10004> E;
array<array<char, 54>, 54> ans;
array<vector<int>, 104> G;
void add(int u, int v, int f){
    E[cnt] = {u, v, f};
    G[u].pb(cnt++);
    E[cnt] = {v, u, 0};
    G[v].pb(cnt++);
}
bool BFS(int s, int t){
    int u;
    queue<int> Q;
    Q.push(s);
    lvl[s] = 1;
    while(!Q.empty()){
        u = Q.front();
        Q.pop();
        for(int i : G[u]){
            auto [o, v, f] = E[i];
            if(lvl[v] || !f) continue;
            lvl[v] = lvl[u] + 1;
            Q.push(v);
        }
    }
    return lvl[t];
}
int DFS(int u, int t, int f){
    if(u == t || !f) return f;
    int tmp, wut = 0;
    for(int &i = P[u]; i < G[u].size(); i++){
        auto &[eu, ev, ef] = E[G[u][i]];
        auto &[bu, bv, bf] = E[G[u][i] ^ 1];
        if(lvl[ev] - lvl[u] != 1) continue;
        tmp = DFS(ev, t, min(f, ef));
        ef -= tmp, bf += tmp;
        f -= tmp, wut += tmp;
    }
    return wut;
}
int flow(int s, int t){
    int wut = 0, tmp;
    while(1){
        for(int &l : lvl) l = 0;
        if(!BFS(s, t)) break;
```

```
        while(1){
            for(int &p : P) p = 0;
            if(tmp = DFS(s, t, inf)) wut += tmp;
            else break;
        }
    }
    return wut;
}
signed main(){
    int n, w, in = 0, out = 0;
    cin >> n;
    for(int i = 1; i <= n; i++){
        cin >> w;
        in += w;
        add(0, i, w);
    }
    for(int i = 1; i <= n; i++){
        cin >> w;
        out += w;
        add(50 + i, 101, w);
    }
    for(int i = 1; i <= n; i++){
        for(int j = 51; j <= 50 + n; j++){
            add(i, j, 1);
        }
    }
    if(flow(0, 101) != in || in != out){
        cout << "-1\n";
        return 0;
    }
    for(int u = 1; u <= n; u++){
        for(int i : G[u]){
            auto [o, v, f] = E[i];
            if(v < 50) continue;
            if(f) ans[u][v - 50] = '.';
            else ans[u][v - 50] = 'X';
        }
    }
    for(int i = 1; i <= n; i++){
        for(int j = 1; j <= n; j++){
            cout << ans[i][j];
        }
        cout << "\n";
    }
    return 0;
}
```

Grid Puzzle II (https://cses.fi/problemset/task/2131)

Min Cost Max Flow

```
#include <bits/stdc++.h>
#define pb push_back
using namespace std;
struct edge{
    int u, v, f, c;
};
const int inf = 1000696969;
int cnt = 0;
array<bool, 104> in;
array<int, 104> dis, B;
array<edge, 10004> E;
array<array<char, 104>, 104> ans;
array<vector<int>, 104> G;
void add(int u, int v, int f, int c){
    E[cnt] = {u, v, f, c};
    G[u].pb(cnt++);
    E[cnt] = {v, u, 0, -c};
    G[v].pb(cnt++);
}
int SPFA(int s, int t){
    int u;
    queue<int> Q;
    Q.push(s);
    dis[s] = 0;
    while(!Q.empty()){
        u = Q.front();
        Q.pop();
        in[u] = 0;
        for(int i : G[u]){
            auto &[o, v, f, c] = E[i];
            if(!f || dis[u] + c >= dis[v]) continue;
            dis[v] = dis[u] + c;
            B[v] = i;
            if(in[v]) continue;
            in[v] = 1;
            Q.push(v);
        }
    }
    return dis[t] == inf? 0 : 1;
}
void back(int u){
    if(!u) return;
    auto &[eu, ev, ef, ec] = E[B[u]];
    auto &[bu, bv, bf, bc] = E[B[u] ^ 1];
    ef--, bf++;
    back(eu);
}
int flow(int s, int t){
    int tmp, wut = 0;
    while(1){
        for(int &d : dis) d = inf;
        for(int &b : B) b = 0;
        for(bool &i : in) i = 0;
```

```
        if(tmp = SPFA(s, t)) wut += tmp;
        else break;
        back(t);
    }
    return wut;
}
signed main(){
    int n, w, out = 0, sum = 0;
    cin >> n;
    for(int i = 1; i <= n; i++){
        cin >> w;
        add(0, i, w, 0);
    }
    for(int i = 51; i <= 50 + n; i++){
        cin >> w;
        out += w;
        add(i, 101, w, 0);
    }
    for(int i = 1; i <= n; i++){
        for(int j = 51; j <= 50 + n; j++){
            cin >> w;
            add(i, j, 1, -w);
        }
    }
    if(flow(0, 101) != out){
        cout << "-1\n";
        return 0;
    }
    for(int u = 1; u <= n; u++){
        for(int i : G[u]){
            auto [o, v, f, c] = E[i];
            if(v < 50) continue;
            if(f) ans[u][v - 50] = '.';
            else{
                ans[u][v - 50] = 'X';
                sum -= c;
            }
        }
    }
    cout << sum << "\n";
    for(int i = 1; i <= n; i++){
        for(int j = 1; j <= n; j++){
            cout << ans[i][j];
        }
        cout << "\n";
    }
    return 0;
}
```

Empty String (https://cses.fi/problemset/task/1080)

DP

```
#include <bits/stdc++.h>
#define int long long
using namespace std;
const int mod = 1e9 + 7;
string S;
array<array<int, 504>, 504> vis;
array<array<int, 504>, 504> c, dp;
int C(int i, int j){
    if(!j || i == j) return 1;
    if(c[i][j]) return c[i][j];
    return c[i][j] = (C(i - 1, j) + C(i - 1, j - 1)) % mod;
}
int DP(int i, int j){
    if(i > j) return 1;
    if(j - i == 1) return S[i] == S[j];
    if((j - i) % 2 == 0) return 0;
    if(vis[i][j]) return dp[i][j];
    vis[i][j] = 1;
    for(int k = i; k <= j; k++){
        if(S[i] != S[k]) continue;
        dp[i][j] += (DP(i + 1, k - 1) * DP(k + 1, j) % mod) * C((j
        dp[i][j] %= mod;
    }
    return dp[i][j];
}
signed main(){
    cin >> S;
    cout << DP(0, S.size() - 1) << "\n";
    return 0;
}
```

Grid Paths (https://cses.fi/problemset/task/1078)

DP

```cpp
#include <bits/stdc++.h>
#define pb push_back
#define int long long
using namespace std;
struct trap{
    int x, y;
};
const int mod = 1e9 + 7;
array<int, 1004> dp;
array<int, 2000004> fac;
vector<trap> T;
bool cmp(trap a, trap b){
    if(a.x == b.x) return a.y < b.y;
    return a.x < b.x;
}
void build(int n){
    fac[0] = 1;
    for(int i = 1; i <= n; i++){
        fac[i] = i * fac[i - 1] % mod;
    }
}
int exp(int x, int k){
    int pro = 1;
    for(int i = 1; i <= k; i <<= 1){
        if(i & k) pro = pro * x % mod;
        x = x * x % mod;
    }
    return pro;
}
int C(int n, int k){
    return (fac[n] * exp(fac[n - k], mod - 2) % mod) * exp(fac[k],
}
int DP(int n){
    for(int i = 0; i <= n; i++){
        auto [x, y] = T[i];
        dp[i] = C(x + y - 2, x - 1);
        for(int j = 0; j < i; j++){
            auto [tx, ty] = T[j];
            if(y < ty) continue;
            dp[i] = (dp[i] - dp[j] * C(x - tx + y - ty, x - tx) % m
        }
    }
    return dp[n];
}
signed main(){
    int n, m, x, y;
    cin >> n >> m;
    build(2 * n);
    for(int i = 1; i <= m; i++){
        cin >> x >> y;
        T.pb({x, y});
    }
    T.pb({n, n});
```

```
54      sort(T.begin(), T.end(), cmp);
55      cout << DP(m) << "\n";
56      return 0;
57  }
```

Bit Substrings (https://cses.fi/problemset/task/2115)

FFT

```cpp
#include <bits/stdc++.h>
#define int long long
#define cpx complex<double>
#define i imag
#define r real
using namespace std;
const int N = 1 << 19;
const double pi = acos(-1);
array<cpx, 1 << 19> A, B, C, X;
cpx ei(double x){
    return {cos(x), sin(x)};
}
void FFT(array<cpx, 1 << 19> &F){
    int n;
    cpx 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, pre = 0, zero = 0, cnt = 0;
    string S;
    cin >> S;
    n = S.size();
    B[n] = {1, 0};
    for(char s : S){
        if(s == '1'){
            pre++;
            zero += cnt * (cnt + 1) / 2;
            cnt = 0;
        }else cnt++;
        A[pre] += 1;
        B[n - pre] += 1;
    }
    zero += cnt * (cnt + 1) / 2;
    for(int i = 0; i < N; i++){
        X[i] = ei(2.0 * 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);
    cout << zero << " ";
    for(int i = n + 1; i <= 2 * n; i++){
        cout << rnd(C[i].r() / (double)N) << " ";
    }
    cout << "\n";
    return 0;
}
```

Reversal Sorting (https://cses.fi/problemset/task/2075)

Treap

```cpp
#include <bits/stdc++.h>
#define pb push_back
using namespace std;
const int inf = 1e9;
struct treap{
    int val, man, siz, pri;
    bool rev;
    treap *lc, *rc;
    treap(int v){
        val = man = v;
        siz = 1;
        pri = rand();
        rev = 0;
        lc = rc = nullptr;
    }
    void pull(){
        man = min(min(lc? lc->man : inf, rc? rc->man : inf), val);
        siz = (lc? lc->siz : 0) + (rc? rc->siz : 0) + 1;
    }
    void push(){
        if(!rev) return;
        swap(lc, rc);
        if(lc) lc->rev ^= 1;
        if(rc) rc->rev ^= 1;
        rev = 0;
    }
    int find(int k){
        push();
        int ls = (lc? lc->siz : 0) + 1;
        if(val == k) return ls;
        if(lc && lc->man == k) return lc->find(k);
        else return rc->find(k) + ls;
    }
};
vector<pair<int, int>> ans;
int size(treap *t){
    return t? t->siz : 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 = nullptr;
        return;
    }
    t->push();
    if(k <= size(t->lc)){
        b = t;
        split(t->lc, a, b->lc, k);
        b->pull();
    }else{
        a = t;
        split(t->rc, a->rc, b, k - size(t->lc) - 1);
        a->pull();
    }
}
signed main(){
    srand(time(NULL));
    int n, x, p;
    treap *t = nullptr, *a = nullptr, *b = nullptr, *c = nullptr;
    cin >> n;
    for(int i = 1; i <= n; i++){
        cin >> x;
        t = merge(t, new treap(x));
    }
    for(int i = 1; i <= n; i++){
        split(t, a, b, i - 1);
        p = b->find(i);
        if(p == 1){
            t = merge(a, b);
            continue;
        }
        ans.pb({i, i + p - 1});
        split(b, b, c, p);
        b->rev ^= 1;
        t = merge(a, merge(b, c));
    }
    cout << ans.size() << "\n";
    for(auto [l, r] : ans) cout << l << " " << r << "\n";
    return 0;
}
```

Counting Reorders (https://cses.fi/problemset/task/2421)

```
嗷嗷待哺
```

Book Shop II (https://cses.fi/problemset/task/1159/)

DP

```cpp
#include <bits/stdc++.h>
#define int long long
using namespace std;
int l, r;
array<int, 104> H, S, K;
array<int, 100004> Q;
array<array<int, 100004>, 2> dp;
int DP(int n, int x){
    int p = 0, ans = 0;
    for(int i = 1; i <= n; i++){
        p ^= 1;
        for(int j = 0; j < H[i]; j++){
            l = 0, r = -1;
            for(int k = j; k <= x; k += H[i]){
                if(l <= r && Q[l] + K[i] * H[i] < k) l++;
                while(l <= r && dp[p ^ 1][k] >= dp[p ^ 1][Q[r]] + (
                Q[++r] = k;
                dp[p][k] = dp[p ^ 1][Q[l]] + (k - Q[l]) / H[i] * S[
                ans = max(ans, dp[p][k]);
            }
        }
    }
    return ans;
}
signed main(){
    int n, x;
    cin >> n >> x;
    for(int i = 1; i <= n; i++) cin >> H[i];
    for(int i = 1; i <= n; i++) cin >> S[i];
    for(int i = 1; i <= n; i++) cin >> K[i];
    cout << DP(n, x) << "\n";
    return 0;
}
```

Network Breakdown (https://cses.fi/problemset/task/1677)

DSU

```cpp
#include <bits/stdc++.h>
#define pb push_back
using namespace std;
int com;
array<int, 100004> DSU;
vector<int> ans;
vector<pair<int, int>> B;
set<pair<int, int>> E;
int find(int u){
    if(DSU[u] == u) 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 run(int k){
    for(auto [a, b] : E){
        onion(a, b);
    }
    ans.pb(com);
    for(auto [a, b] : B){
        onion(a, b);
        ans.pb(com);
    }
    ans.pop_back();
}
signed main(){
    int n, m, k, a, b;
    cin >> n >> m >> k;
    com = n;
    for(int i = 1; i <= n; i++) DSU[i] = i;
    while(m--){
        cin >> a >> b;
        if(a > b) swap(a, b);
        E.insert({a, b});
    }
    for(int i = 0; i < k; i++){
        cin >> a >> b;
        if(a > b) swap(a, b);
        E.erase({a, b});
        B.pb({a, b});
    }
    reverse(B.begin(), B.end());
    run(k);
    reverse(ans.begin(), ans.end());
    for(int c : ans) cout << c << " ";
    cout << "\n";
    return 0;
}
```

Visiting Cities (https://cses.fi/problemset/task/1203)

Dijkstra Dominator Tree

```cpp
#include <bits/stdc++.h>
#define pb push_back
#define int long long
using namespace std;
struct edge{
    int v, w;
};
struct cmp{
    bool operator()(edge a, edge b){
        return a.w > b.w;
    }
};
int k = 0;
array<int, 100004> dis, P, rev, L, sdom, idom, pre, DSU;
array<vector<int>, 100004> D, R, buk;
array<vector<edge>, 100004> G;
vector<int> ans;
int find(int u, int x = 0){
    if(u == DSU[u]) return x? -1 : u;
    int v = find(DSU[u], x + 1);
    if(v < 0) return u;
    if(sdom[L[DSU[u]]] < sdom[L[u]]) L[u] = L[DSU[u]];
    DSU[u] = v;
    return x? v : L[u];
}
void onion(int u, int v){
    DSU[v] = u;
}
void run(int s, int n){
    priority_queue<edge, vector<edge>, cmp> Q;
    Q.push({s, 1});
    while(!Q.empty()){
        auto [u, d] = Q.top();
        Q.pop();
        if(dis[u]) continue;
        dis[u] = d;
        for(auto [v, w] : G[u]){
            Q.push({v, d + w});
        }
    }
    for(int u = 1; u <= n; u++){
        for(auto [v, w] : G[u]){
            if(dis[u] + w == dis[v]) D[u].pb(v);
        }
    }
}
void DFS(int u){
    P[u] = ++k, rev[k] = u, L[k] = k, sdom[k] = k, DSU[k] = k;
    for(int v : D[u]){
        if(!P[v]){
            DFS(v);
            pre[P[v]] = P[u];
        }
```

```
        R[P[v]].pb(P[u]);
    }
}
void DOM(int n){
    int w;
    for(int u = n; u; u--){
        for(int v : R[u]){
            sdom[u] = min(sdom[u], sdom[find(v)]);
        }
        if(u > 1) buk[sdom[u]].pb(u);
        for(int v : buk[u]){
            w = find(v);
            if(sdom[v] == sdom[w]) idom[v] = sdom[v];
            else idom[v] = w;
        }
        onion(pre[u], u);
    }
    for(int u = 2; u <= n; u++){
        if(sdom[u] != idom[u]) idom[u] = idom[idom[u]];
    }
}
void back(int u){
    ans.pb(rev[u]);
    if(u == 1) return;
    back(idom[u]);
}
signed main(){
    int n, m, a, b, c;
    cin >> n >> m;
    while(m--){
        cin >> a >> b >> c;
        G[a].pb({b, c});
    }
    run(1, n);
    DFS(1);
    DOM(n);
    back(P[n]);
    sort(ans.begin(), ans.end());
    cout << ans.size() << "\n";
    for(int u : ans) cout << u << " ";
    cout << "\n";
    return 0;
}
```

Missing Coin Sum Queries (https://cses.fi/problemset/task/2184)

Doubling RMQ

```cpp
#include <bits/stdc++.h>
#define pb push_back
#define int long long
using namespace std;
struct ooo{
    int l, r, t;
};
array<int, 200004> X, ans, P;
array<int, 400004> S;
array<array<int, 200004>, 20> T;
vector<ooo> Q;
int highbit(int x){
    int p = 0;
    for(int i = 4; i >= 0; i--){
        if(1 << (p + (1 << i)) <= x) p += (1 << i);
    }
    return p;
}
void build(int n){
    for(int i = 1; 1 << i <= n; i++){
        for(int j = 1; j <= n; j++){
            T[i][j] = min(T[i - 1][j], T[i - 1][j + (1 << (i - 1))]
        }
    }
}
void find(int n){
    int h;
    for(int k = 1, p = 1; k < 1 << 30; k <<= 1, p = 1){
        S[1] = 0, T[0][1] = 1ll << 60;
        for(int i = 1; i <= n; i++){
            if(X[i] >= k && X[i] < k << 1){
                P[i] = ++p;
                S[p] = X[i] + S[p - 1];
                T[0][p] = X[i];
                p++;
            }else P[i] = p;
            S[p] = S[p - 1], T[0][p] = 1ll << 60;
        }
        build(p);
        for(auto [l, r, t] : Q){
            l = P[l], r = P[r], h = highbit(r - l + 1);
            if(ans[t] >= min(T[h][l], T[h][r + 1 - (1 << h)])){
                ans[t] += S[r] - S[l - 1];
            }
        }
    }
}
signed main(){
    int n, q, l, r;
    cin >> n >> q;
    for(int i = 1; i <= n; i++) cin >> X[i];
    for(int i = 1; i <= q; i++){
        cin >> l >> r;
```

```
            Q.pb({l, r, i});
            ans[i] = 1;
        }
        find(n);
        for(int i = 1; i <= q; i++) cout << ans[i] << "\n";
        return 0;
    }
```

Number Grid (https://cses.fi/problemset/task/1157)

通靈

```
#include <bits/stdc++.h>
using namespace std;
signed main(){
    int x, y;
    cin >> x >> y;
    cout << ((x - 1) ^ (y - 1)) << "\n";
    return 0;
}
```

Maximum Building II (https://cses.fi/problemset/task/1148/)

差分

```cpp
#include <bits/stdc++.h>
#define int long long
using namespace std;
array<int, 1004> H;
array<array<char, 1004>, 1004> G;
array<array<int, 1004>, 1004> cnt;
void run(int n, int m){
    int now;
    for(int i = 1; i <= n; i++){
        stack<pair<int, int>> S;
        S.push({0, 0});
        for(int j = 1; j <= m; j++){
            now = j;
            if(G[i][j] == '*') H[j] = 0;
            else H[j]++;
            while(!S.empty()){
                auto [h, p] = S.top();
                if(H[j] < h){
                    cnt[h][j - p]++;
                    S.pop();
                    now = p;
                    auto [h2, p2] = S.top();
                    cnt[max(h2, H[j])][j - p]--;
                }else{
                    break;
                }
            }
            S.push({H[j], now});
        }
    }
    for(int i = n; i; i--){
        for(int j = m; j; j--){
            cnt[i][j] += cnt[i][j + 1];
        }
    }
    for(int i = n; i; i--){
        for(int j = m; j; j--){
            cnt[i][j] += cnt[i + 1][j] + cnt[i][j + 1] - cnt[i + 1]
        }
    }
}
signed main(){
    int n, m;
    cin >> n >> m;
    for(int i = 1; i <= n; i++){
        for(int j = 1; j <= m; j++){
            cin >> G[i][j];
        }
        G[i][m + 1] = '*';
    }
    run(n, m + 1);
    for(int i = 1; i <= n; i++){
        for(int j = 1; j <= m; j++){
```

```
            cout << cnt[i][j] << " ";
        }
        cout << "\n";
    }
    return 0;
}
```

Filling Trominos (https://cses.fi/problemset/task/2423)

```
嗷嗷待哺
```

Stick Divisions (https://cses.fi/problemset/task/1161)

Greedy

```
#include <bits/stdc++.h>
#define int long long
using namespace std;
array<int, 200004> L;
int div(int n){
    int a, b, sum = 0;
    priority_queue<int, vector<int>, greater<int>> Q;
    for(int i = 0; i < n; i++) Q.push(L[i]);
    while(Q.size() > 1){
        a = Q.top(), Q.pop();
        b = Q.top(), Q.pop();
        sum += a + b;
        Q.push(a + b);
    }
    return sum;
}
signed main(){
    int x, n;
    cin >> x >> n;
    for(int i = 0; i < n; i++){
        cin >> L[i];
    }
    cout << div(n) << "\n";
    return 0;
}
```

Coding Company (https://cses.fi/problemset/task/1665)

DP

```cpp
#include <bits/stdc++.h>
#define int long long
using namespace std;
const int mod = 1e9 + 7;
array<int, 104> T;
array<array<array<int, 20004>, 104>, 2> dp;
int DP(int n, int x, int sum){
    int p = 0, ans = 0;
    dp[0][0][sum] = 1;
    for(int i = 1; i <= n; i++){
        p ^= 1;
        for(int j = 0; j <= i; j++){
            for(int k = -sum; k <= sum; k++){
                dp[p][j][k + sum] = 0;
                if(j && k + T[i] <= sum) dp[p][j][k + sum] += dp[p
                dp[p][j][k + sum] += (j + 1) * dp[p ^ 1][j][k + sum
                if(k - T[i] >= -sum) dp[p][j][k + sum] += (j + 1) *
                dp[p][j][k + sum] %= mod;
            }
        }
    }
    for(int i = 0; i <= x; i++){
        ans = (ans + dp[n & 1][0][i + sum]) % mod;
    }
    return ans;
}
signed main(){
    int n, x, sum = 0;
    cin >> n >> x;
    for(int i = 1; i <= n; i++){
        cin >> T[i];
        sum += T[i];
    }
    sort(T.begin() + 1, T.begin() + n + 1);
    cout << DP(n, x, sum) << "\n";
    return 0;
}
```

Flight Route Requests (https://cses.fi/problemset/task/1699)

Topological Sort

```cpp
#include <bits/stdc++.h>
#define pb push_back
using namespace std;
bitset<100004> vis;
array<int, 100004> in;
array<vector<pair<int, int>>, 100004> G;
vector<int> V;
void DFS(int u){
    if(vis[u]) return;
    vis[u] = 1;
    V.pb(u);
    for(auto [v, d] : G[u]){
        if(d) in[v]++;
        DFS(v);
    }
}
int topu(int u){
    if(vis[u]) return 0;
    int nc = 0;
    V.clear();
    DFS(u);
    queue<int> Q;
    for(int v : V){
        if(!in[v]) Q.push(v);
    }
    while(!Q.empty()){
        u = Q.front();
        Q.pop();
        nc++;
        for(auto [v, d] : G[u]){
            if(!d) continue;
            in[v]--;
            if(!in[v]) Q.push(v);
        }
    }
    return nc == V.size()? V.size() - 1 : V.size();
}
signed main(){
    int n, m, a, b, edge = 0;
    cin >> n >> m;
    while(m--){
        cin >> a >> b;
        G[a].pb({b, 1});
        G[b].pb({a, 0});
    }
    for(int i = 1; i <= n; i++){
        edge += topu(i);
    }
    cout << edge << "\n";
    return 0;
}
```

Two Stacks Sorting (https://cses.fi/problemset/task/2402)

```
1 嗷嗷待哺
```

Tree Isomorphism II (https://cses.fi/problemset/task/1701)

`Tree` `Centriod` `Hash`

```cpp
#include <bits/stdc++.h>
#define int long long
#define pb push_back
using namespace std;
const int k = 41, mod = 1000696969;
int n;
array<int, 100004> H, S1, S2;
array<vector<int>, 100004> T1, T2;
vector<int> C1, C2;
void build(){
    H[0] = 1;
    for(int i = 1; i < 100004; i++){
        H[i] = k * H[i - 1] % mod;
    }
}
int DFS(array<vector<int>, 100004> &T, vector<int> &C, int u, int p
    int tmp, s = 1;
    bool cen = 1;
    for(int v : T[u]){
        if(v == pre) continue;
        tmp = DFS(T, C, v, u);
        if(2 * tmp > n) cen = 0;
        s += tmp;
    }
    if(2 * s < n) cen = 0;
    if(cen) C.pb(u);
    return s;
}
int HAS(array<vector<int>, 100004> &T, array<int, 100004> &S, int u
    int sum = 0;
    S[u] = 0;
    vector<pair<int, int>> has;
    for(int v : T[u]){
        if(v == pre) continue;
        has.pb({HAS(T, S, v, u), S[v]});
    }
    sort(has.begin(), has.end());
    for(auto [h, s] : has){
        sum = (sum + H[S[u]] * h) % mod;
        S[u] += s;
    }
    sum = (sum + H[S[u]] * (S[u] + 1)) % mod;
    S[u]++;
    return sum;
}
signed main(){
    int t, a, b;
    bool ok;
    cin >> t;
    build();
    while(t--){
        cin >> n;
        ok = 0;
```

```
54          C1.clear(), C2.clear();
55          for(int i = 1; i <= n; i++){
56              T1[i].clear();
57              T2[i].clear();
58          }
59          for(int i = 1; i < n; i++){
60              cin >> a >> b;
61              T1[a].pb(b);
62              T1[b].pb(a);
63          }
64          for(int i = 1; i < n; i++){
65              cin >> a >> b;
66              T2[a].pb(b);
67              T2[b].pb(a);
68          }
69          DFS(T1, C1, 1, 0);
70          DFS(T2, C2, 1, 0);
71          for(int c1 : C1){
72              for(int c2 : C2){
73                  if(HAS(T1, S1, c1, 0) == HAS(T2, S2, c2, 0)) ok = 1
74              }
75          }
76          if(C1.size() != C2.size()) ok = 0;
77          cout << (ok? "YES\n" : "NO\n");
78      }
79      return 0;
80  }
```

Forbidden Cities (https://cses.fi/problemset/task/1705)

圓方樹　LCA　樹壓平

```cpp
#include <bits/stdc++.h>
#define int long long
#define pb push_back
using namespace std;
struct edge{
    int v, t;
};
int k = 0, r = 0, o = 0;
bitset<200004> nee, vis;
array<int, 100004> P, low, bcc;
array<int, 200004> in, out, ecc;
array<int, 400004> BIT;
array<array<int, 20>, 200004> A;
array<vector<int>, 200004> T;
array<vector<edge>, 100004> G;
stack<int> S;
set<int> E;
void update(int p, int x){
    for(; p < 400004; p += p & -p) BIT[p] += x;
}
int query(int p){
    int sum = 0;
    for(; p; p -= p & -p) sum += BIT[p];
    return sum;
}
int hesh(int u, int v){
    if(u > v) swap(u, v);
    return u * 100001 + v;
}
void DFST(int u, int pre){
    int cnt = 0, s;
    P[u] = low[u] = ++r;
    for(auto [v, t] : G[u]){
        if(v == pre) continue;
        if(!vis[t]){
            S.push(t);
            vis[t] = 1;
        }
        if(P[v]) low[u] = min(low[u], P[v]);
        else{
            cnt++;
            DFST(v, u);
            low[u] = min(low[u], low[v]);
            if(low[v] >= P[u]){
                k++, nee[u] = 1;
                while(!S.empty()){
                    s = S.top(), S.pop();
                    ecc[s] = k;
                    if(s == t) break;
                }
            }
        }
    }
```

```
    if(u == 1 && cnt < 2) nee[1] = 0;
}
void DFS(int u, int pre){
    in[u] = ++o;
    for(int v : T[u]){
        if(v == pre) continue;
        DFS(v, u);
        A[v][0] = u;
    }
    out[u] = ++o;
}
void DABO(int n){
    in[0] = 0, out[0] = 1 << 20;
    for(int j = 1; j < 20; j++){
        for(int i = 1; i <= n; i++){
            A[i][j] = A[A[i][j - 1]][j - 1];
        }
    }
}
int LCA(int a, int b){
    for(int i = 19; i >= 0; i--){
        if(in[A[a][i]] > in[b] || out[A[a][i]] < out[b]) a = A[a][
    }
    if(in[a] > in[b] || out[a] < out[b]) a = A[a][0];
    return a;
}
signed main(){
    int n, m, q, a, b, c, lca, ans;
    cin >> n >> m >> q;
    for(int i = 1; i <= m; i++){
        cin >> a >> b;
        if(E.find(hesh(a, b)) != E.end() || a == b) continue;
        E.insert(hesh(a, b));
        G[a].pb({b, i});
        G[b].pb({a, i});
    }
    DFST(1, 0);
    for(int i = 1; i <= n; i++){
        if(nee[i]) bcc[i] = ++k;
        else bcc[i] = ecc[G[i][0].t];
    }
    for(int u = 1; u <= n; u++){
        if(!nee[u]) continue;
        for(auto [v, t] : G[u]){
            T[bcc[u]].pb(ecc[t]);
            T[ecc[t]].pb(bcc[u]);
        }
    }
    for(int i = 1; i <= k; i++){
        sort(T[i].begin(), T[i].end());
        T[i].erase(unique(T[i].begin(), T[i].end()), T[i].end());
    }
    DFS(1, 0);
    DABO(k);
```

```
    DABO(k);
    while(q--){
        cin >> a >> b >> c;
        if(a == c || b == c){
            cout << "NO\n";
            continue;
        }else if(!nee[c]){
            cout << "YES\n";
            continue;
        }
        a = bcc[a], b = bcc[b], c = bcc[c];
        update(in[c], 1), update(out[c], -1);
        lca = LCA(a, b);
        if(in[a] > in[b]) swap(a, b);
        if(c == lca) ans = 1;
        else if(a == lca) ans = query(in[b]) - query(in[a] - 1);
        else ans = query(in[b]) - query(out[a] - 1);
        update(in[c], -1), update(out[c], 1);
        cout << (ans? "NO\n" : "YES\n");
    }
    return 0;
}
```

Area of Rectangles (https://cses.fi/problemset/task/1741)

Segment Tree

```cpp
#include <bits/stdc++.h>
#define pb push_back
#define int long long
#define mid ((l + r) >> 1)
#define lc (p << 1)
#define rc ((p << 1) | 1)
using namespace std;
struct ooo{
    int x, l, r, v;
};
const int inf = 1e6;
array<int, 8000004> man, tag, cnt;
vector<ooo> Q;
bool cmp(ooo a, ooo b){
    return a.x < b.x;
}
void pull(int p){
    man[p] = min(man[lc], man[rc]);
    if(man[lc] < man[rc]) cnt[p] = cnt[lc];
    else if(man[rc] < man[lc]) cnt[p] = cnt[rc];
    else cnt[p] = cnt[lc] + cnt[rc];
}
void push(int p){
    man[lc] += tag[p];
    man[rc] += tag[p];
    tag[lc] += tag[p];
    tag[rc] += tag[p];
    tag[p] = 0;
}
void build(int p, int l, int r){
    if(l == r){
        cnt[p] = 1;
        return;
    }
    build(lc, l, mid);
    build(rc, mid + 1, r);
    pull(p);
}
void update(int p, int l, int r, int ql, int qr, int x){
    if(ql > r || qr < l) return;
    if(ql <= l && qr >= r){
        man[p] += x;
        tag[p] += x;
        return;
    }
    push(p);
    update(lc, l, mid, ql, qr, x);
    update(rc, mid + 1, r, ql, qr, x);
    pull(p);
}
signed main(){
    int n, x1, y1, x2, y2, p = 0, sum = 0;
    cin >> n;
```

```
    for(int i = 1; i <= n; i++){
        cin >> x1 >> y1 >> x2 >> y2;
        Q.pb({x1, y1, y2 - 1, 1});
        Q.pb({x2, y1, y2 - 1, -1});
    }
    sort(Q.begin(), Q.end(), cmp);
    build(1, -inf, inf);
    for(int i = -inf; i < inf; i++){
        while(p < Q.size() && Q[p].x == i){
            auto [x, l, r, v] = Q[p++];
            update(1, -inf, inf, l, r, v);
        }
        sum += 2 * inf + 1 - cnt[1];
    }
    cout << sum << "\n";
    return 0;
}
```

Grid Completion (https://cses.fi/problemset/task/2429)

```
嗷嗷待哺
```

Creating Offices (https://cses.fi/problemset/task/1752)

重心剖分

```cpp
#include <bits/stdc++.h>
#define pb push_back
using namespace std;
array<int, 200004> S, M, D, F, N;
array<array<int, 20>, 200004> dis;
array<vector<int>, 200004> T;
vector<int> V, ans;
vector<pair<int, int>> Q;
int DFS(int u){
    if(S[u]) return 0;
    int s;
    S[u] = 1, V.pb(u);
    for(int v : T[u]){
        s = DFS(v);
        M[u] = max(M[u], s), S[u] += s;
    }
    return S[u];
}
void walk(int u, int dep, int s){
    if(S[u]) return;
    S[u] = 1, dis[u][dep] = s;
    for(int v : T[u]){
        walk(v, dep, s + 1);
    }
    S[u] = 0;
}
int CUT(int u, int dep){
    V.clear();
    DFS(u);
    int cen, n = V.size();
    for(int v : V){
        if(2 * M[v] <= n && 2 * S[v] >= n) cen = v;
        S[v] = M[v] = 0;
    }
    walk(cen, dep, 0);
    D[cen] = dep, S[cen] = 1;
    for(int v : T[cen]){
        if(!S[v]) F[CUT(v, dep + 1)] = cen;
    }
    return cen;
}
void RUN(int u, int pre, int dep){
    Q.pb({dep, u});
    for(int v : T[u]){
        if(v == pre) continue;
        RUN(v, u, dep + 1);
    }
}
void update(int u, int v){
    if(!u) return;
    N[u] = min(N[u], dis[v][D[u]]);
    update(F[u], v);
}
```

```
int query(int u, int v){
    if(!u) return 1 << 20;
    return min(N[u] + dis[v][D[u]], query(F[u], v));
}
signed main(){
    cin.tie(0), cout.tie(0), ios::sync_with_stdio(0);
    int n, d, a, b;
    cin >> n >> d;
    for(int i = 1; i < n; i++){
        cin >> a >> b;
        T[a].pb(b);
        T[b].pb(a);
    }
    CUT(1, 0);
    for(int &s : S) s = 0;
    for(int &s : N) s = 1 << 20;
    RUN(1, 0, 0);
    sort(Q.begin(), Q.end());
    reverse(Q.begin(), Q.end());
    for(auto [s, u] : Q){
        if(query(u, u) < d) continue;
        ans.pb(u);
        update(u, u);
    }
    cout << ans.size() << "\n";
    for(int u : ans) cout << u << " ";
    cout << "\n";
    return 0;
}
```

Permutations II (https://cses.fi/problemset/task/1075)

DP

```cpp
#include <bits/stdc++.h>
#define int long long
using namespace std;
const int mod = 1e9 + 7;
array<array<array<int, 2>, 1004>, 1004> dp;
int DP(int n){
    dp[1][0][0] = 1;
    dp[2][1][1] = 2;
    for(int i = 3; i <= n; i++){
        for(int j = 0; j < i; j++){
            dp[i][j][0] += (i - j - 2) * dp[i - 1][j][0];
            dp[i][j][0] += (i - j - 1) * dp[i - 1][j][1];
            dp[i][j][0] += (j + 1) * dp[i - 1][j + 1][0];
            dp[i][j][0] += j * dp[i - 1][j + 1][1];
            if(j) dp[i][j][1] += 2 * dp[i - 1][j - 1][0];
            dp[i][j][1] += dp[i - 1][j][1];
            if(j) dp[i][j][1] += dp[i - 1][j - 1][1];
            dp[i][j][0] %= mod, dp[i][j][1] %= mod;
        }
    }
    return dp[n][0][0];
}
signed main(){
    int n;
    cin >> n;
    cout << DP(n) << "\n";
    return 0;
}
```

Functional Graph Distribution (https://cses.fi/problemset/task/2415)

Caley's Formula

```cpp
#include <bits/stdc++.h>
#define int long long
using namespace std;
const int mod = 1e9 + 7;
array<int, 5004> N;
array<array<int, 5004>, 5004> C, S;
void build(int n){
    N[0] = S[0][0] = C[0][0] = 1;
    for(int i = 1; i <= n; i++){
        N[i] = n * N[i - 1] % mod;
        C[i][0] = 1;
        for(int j = 1; j <= i; j++){
            C[i][j] = (C[i - 1][j] + C[i - 1][j - 1]) % mod;
            S[i][j] = ((i - 1) * S[i - 1][j] + S[i - 1][j - 1]) % m
        }
    }
}
signed main(){
    int n, sum = 0;
    cin >> n;
    build(n);
    for(int i = 1; i <= n; i++, sum = 0){
        for(int j = 1; j < n; j++){
            sum = (sum + ((C[n][j] * S[j][i]) % mod) * ((j * N[n -
        }
        sum = (sum + S[n][i]) % mod;
        cout << sum << "\n";
    }
    return 0;
}
```

New Flight Routes (https://cses.fi/problemset/task/1685)

SCC Maximal Flow

```cpp
#include <bits/stdc++.h>
#define pb push_back
using namespace std;
const int t = 100001;
int k = 0;
bitset<100004> vis, F;
array<int, 100004> in, out, scc, S;
array<vector<int>, 100004> G, R;
array<vector<pair<int, int>>, 100004> D;
vector<int> ord, I, O, A, B;
vector<pair<int, int>> ans;
void BFS(int u){
    if(vis[u]) return;
    vis[u] = 1;
    for(int v : R[u]) BFS(v);
    ord.pb(u);
}
void DFS(int u){
    if(scc[u]) return;
    scc[u] = k;
    for(int v : G[u]) DFS(v);
}
bool flow(int u){
    if(u == t) return 1;
    for(auto &[v, f] : D[u]){
        if(f) continue;
        f = 1;
        if(flow(v)){
            if(!in[u]) I.pb(u), F[u] = 1;
            if(!out[u]) O.pb(u), F[u] = 1;
            return 1;
        }
    }
    return 0;
}
signed main(){
    int n, m, a, b, s;
    cin >> n >> m;
    while(m--){
        cin >> a >> b;
        G[a].pb(b);
        R[b].pb(a);
    }
    for(int i = 1; i <= n; i++) BFS(i);
    reverse(ord.begin(), ord.end());
    for(int u : ord){
        if(!scc[u]) k++;
        DFS(u);
    }
    if(k == 1){
        cout << "0\n";
        return 0;
    }
```

```
    for(int u = 1; u <= n; u++){
        S[scc[u]] = u;
        for(int v : G[u]){
            if(scc[v] == scc[u]) continue;
            D[scc[u]].pb({scc[v], 0});
        }
    }
    for(int i = 1; i <= k; i++){
        D[i].erase(unique(D[i].begin(), D[i].end()), D[i].end());
        for(auto [j, f] : D[i]){
            in[j]++, out[i]++;
        }
    }
    for(int i = 1; i <= k; i++){
        if(!out[i]) D[i].pb({t, 0});
    }
    for(int i = 1; i <= k; i++){
        if(!in[i]) flow(i), a = i;
        if(!out[i]) b = i;
    }
    for(int i = 1; i <= k; i++){
        if(F[i]) continue;
        if(!in[i]) A.pb(i);
        if(!out[i]) B.pb(i);
    }
    for(int i = 0; i < I.size() - 1; i++){
        ans.pb({O[i], I[i + 1]});
    }
    if(I.size()) ans.pb({O.back(), I[0]});
    for(int i = 0; i < min(A.size(), B.size()); i++){
        ans.pb({B[i], A[i]});
    }
    if(A.size() > B.size()){
        for(int i = B.size(); i < A.size(); i++){
            ans.pb({b, A[i]});
        }
    }else{
        for(int i = A.size(); i < B.size(); i++){
            ans.pb({B[i], a});
        }
    }
    cout << ans.size() << "\n";
    for(auto [u, v] : ans) cout << S[u] << " " << S[v] << "\n";
    return 0;
}
```

Grid Path Construction (https://cses.fi/problemset/task/2418)

```
嗷嗷待哺
```