

CSES PLAN IV

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

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) (<https://hackmd.io/@peienwu/cses#CSES-%E8%A3%9C%E5%AE%8C%E8%A8%88%E7%95%AB>)

目錄

[CSES PLAN I](https://hackmd.io/@thanksone/CSESPLANI) (<https://hackmd.io/@thanksone/CSESPLANI>)

- Introductory Problems
- Sorting and Searching
- Dynamic Programming
- Graph Algorithms

[CSES PLAN II](https://hackmd.io/@thanksone/CSESPLANII) (<https://hackmd.io/@thanksone/CSESPLANII>)

- Range Queries
- Tree Algorithms
- Mathematics
- String Algorithms
- Geometry

[CSES PLAN III](https://hackmd.io/@thanksone/CSESPLANIII) (<https://hackmd.io/@thanksone/CSESPLANIII>)

- Advanced Techniques

[CSES PLAN IV](https://hackmd.io/@thanksone/CSESPLANIV) (<https://hackmd.io/@thanksone/CSESPLANIV>)

- Additional Problems

Additional Problems

[Shortest Subsequence](https://cses.fi/problemset/task/1087) (<https://cses.fi/problemset/task/1087>)

Greedy

```
1 #include <bits/stdc++.h>
2 using namespace std;
3 array<char, 4> C = {'A', 'C', 'G', 'T'};
4 array<array<int, 1000004>, 4> nxt;
5 void build(string &DNA){
6     int n = DNA.size();
7     for(int i = 0; i < 4; i++) nxt[i][n] = n;
8     for(int i = n - 1; i >= 0; i--){
9         for(int j = 0; j < 4; j++){
10            if(DNA[i] == C[j]) nxt[j][i] = i;
11            else nxt[j][i] = nxt[j][i + 1];
12        }
13    }
14 }
15 void print(string &DNA){
16     int n = DNA.size(), p, now = 0;
17     char c;
18     for(int i = 0; i < n; i++){
19         p = 0;
20         for(int j = 0; j < 4; j++){
21             if(nxt[j][now] > p){
22                 p = nxt[j][now];
23                 c = C[j];
24             }
25         }
26         cout << c;
27         now = p + 1;
28         if(now > n) return;
29     }
30 }
31 signed main(){
32     string DNA;
33     cin >> DNA;
34     build(DNA);
35     print(DNA);
36     return 0;
37 }
```

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

爆搜

```
1 #include <bits/stdc++.h>
2 #define int long long
3 using namespace std;
4 int cal(int n){
5     int ans = 0;
6     for(int i = 1ll << 50; i > 0; i >>= 1){
7         if(n <= i) continue;
8         ans += (n / (i << 1)) * i;
9         ans += (n % (i << 1)) - min(i, n % (i << 1));
10    }
11    return ans;
12 }
13 signed main(){
14     int n;
15     cin >> n;
16     cout << cal(n + 1) << "\n";
17     return 0;
18 }
```

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

BFS


```
1 #include <bits/stdc++.h>
2 using namespace std;
3 struct pos{
4     int u, s;
5 };
6 int T = 0;
7 array<int, 10> P;
8 bitset<387420489> vis;
9 int swap(int u, int i, int j){
10     int a, b;
11     a = (u / P[i]) % 9;
12     b = (u / P[j]) % 9;
13     u += (b - a) * P[i];
14     u += (a - b) * P[j];
15     return u;
16 }
17 int bfs(int S){
18     int v;
19     queue<pos> Q;
20     Q.push({S, 0});
21     vis[S] = 1;
22     while(1){
23         auto [u, s] = Q.front();
24         Q.pop();
25         if(u == T) return s;
26         for(int i = 0; i < 9; i++){
27             if(i % 3 < 2){
28                 v = swap(u, i, i + 1);
29                 if(!vis[v]){
30                     Q.push({v, s + 1});
31                     vis[v] = 1;
32                 }
33             }
34             if(i + 3 < 9){
35                 v = swap(u, i, i + 3);
36                 if(!vis[v]){
37                     Q.push({v, s + 1});
38                     vis[v] = 1;
39                 }
40             }
41         }
42     }
43 }
44 signed main(){
45     int x, G = 0;
46     P[0] = 1;
47     for(int i = 1; i < 10; i++) P[i] = 9 * P[i - 1];
48     for(int i = 0; i < 9; i++) T += i * P[i];
49     for(int i = 0; i < 9; i++){
50         cin >> x;
51         G += (x - 1) * P[i];
52     }
53     cout << bfs(G) << "\n";
}
```

```
54     return 0;  
55 }
```

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

Priority Queue

```
1 #include <bits/stdc++.h>  
2 using namespace std;  
3 array<int, 200004> P, L;  
4 void run(int n){  
5     priority_queue<int, vector<int>, greater<int>> Q;  
6     for(int i = 1; i <= n; i++){  
7         if(!L[i]) Q.push(i);  
8     }  
9     for(int i = 1; i <= n - 2; i++){  
10        cout << Q.top() << " " << P[i] << "\n";  
11        Q.pop();  
12        if(i == L[P[i]]) Q.push(P[i]);  
13    }  
14    cout << Q.top() << " " << P[n - 2] << "\n";  
15 }  
16 signed main(){  
17     int n;  
18     cin >> n;  
19     for(int i = 1; i <= n - 2; i++){  
20         cin >> P[i];  
21         L[P[i]] = i;  
22     }  
23     L[P[n - 2]] = n;  
24     run(n);  
25     return 0;  
26 }
```

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

DFS Tree

```
1 #include <bits/stdc++.h>
2 #define pb push_back
3 using namespace std;
4 struct edge{
5     int v, d;
6 };
7 bitset<100004> vis;
8 array<int, 100004> dep;
9 array<vector<edge>, 100004> G;
10 void dfs(int u, int pre, int d){
11     vis[u] = 1;
12     dep[u] = d;
13     for(auto &[v, t] : G[u]){
14         if(vis[v]){
15             if(dep[v] > d) t = 1;
16         }else{
17             dfs(v, u, d + 1);
18             t = 1;
19         }
20     }
21 }
22 signed main(){
23     int n, m, a, b;
24     cin >> n >> m;
25     while(m--){
26         cin >> a >> b;
27         G[a].pb({b, 0});
28         G[b].pb({a, 0});
29     }
30     for(int i = 1; i <= n; i++){
31         if(!vis[i]) dfs(i, 0, 1);
32     }
33     for(int i = 1; i <= n; i++){
34         for(auto [v, t] : G[i]){
35             if(t) cout << i << " " << v << "\n";
36         }
37     }
38     return 0;
39 }
```

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

SCC DFS Tree


```
1 #include <bits/stdc++.h>
2 #define pb push_back
3 using namespace std;
4 struct edge{
5     int v, d;
6 };
7 int k = 0;
8 bitset<100004> vis;
9 array<int, 100004> dep, scc;
10 array<vector<edge>, 100004> G;
11 stack<int> out;
12 void dfst(int u, int pre, int d){
13     dep[u] = d;
14     for(auto &[v, t] : G[u]){
15         if(v == pre) continue;
16         if(dep[v]){
17             if(dep[v] < d) t = 1;
18         }else{
19             dfst(v, u, d + 1);
20             t = 1;
21         }
22     }
23 }
24 void bfs(int u){
25     if(vis[u]) return;
26     vis[u] = 1;
27     for(auto [v, t] : G[u]){
28         if(!t) bfs(v);
29     }
30     out.push(u);
31 }
32 void dfs(int u){
33     if(scc[u]) return;
34     scc[u] = k;
35     for(auto [v, t] : G[u]){
36         if(t) dfs(v);
37     }
38 }
39 signed main(){
40     int n, m, a, b, u;
41     cin >> n >> m;
42     while(m--){
43         cin >> a >> b;
44         G[a].pb({b, 0});
45         G[b].pb({a, 0});
46     }
47     for(int i = 1; i <= n; i++){
48         if(!dep[i]) dfst(i, 0, 1);
49     }
50     for(int i = 1; i <= n; i++) bfs(i);
51     while(!out.empty()){
52         u = out.top();
53         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


```
1 #include <bits/stdc++.h>
2 #define pb push_back
3 using namespace std;
4 struct edge{
5     int v, d;
6 };
7 int cnt = 0;
8 array<bool, 100004> odd;
9 array<int, 100004> dep;
10 array<edge, 400004> E;
11 array<vector<int>, 100004> G;
12 void add(int a, int b){
13     G[a].pb(cnt);
14     E[cnt++] = {b, 0};
15     G[b].pb(cnt);
16     E[cnt++] = {a, 0};
17 }
18 void dfs(int u, int pre, int d){
19     dep[u] = d;
20     for(int i : G[u]){
21         auto &[v, t] = E[i];
22         if(v == pre) continue;
23         if(dep[v]){
24             if(dep[v] < d){
25                 t = 1;
26                 odd[u] ^= 1;
27             }
28         }else{
29             dfs(v, u, d + 1);
30             if(!E[i ^ 1].d){
31                 t = 1;
32                 odd[u] ^= 1;
33             }
34         }
35     }
36     if(odd[u]){
37         for(int i : G[u]){
38             auto &[v, t] = E[i];
39             if(v == pre){
40                 t = 1;
41                 odd[u] ^= 1;
42             }
43         }
44     }
45 }
46 signed main(){
47     int n, m, a, b;
48     cin >> n >> m;
49     while(m--){
50         cin >> a >> b;
51         add(a, b);
52     }
53     for(int i = 1; i <= n; i++) {
```

```

54         if(!depl[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

```
1 #include <bits/stdc++.h>
2 #define int long long
3 using namespace std;
4 array<int, 200004> K;
5 stack<pair<int, int>> F;
6 int run(int n){
7     int ans = 0, p;
8     for(int i = 1; i <= n; i++){
9         p = i;
10        while(!F.empty()){
11            auto [l, h] = F.top();
12            if(K[i] < h){
13                ans = max(ans, h * (i - l));
14                F.pop();
15                p = l;
16            }
17            else break;
18        }
19        F.push({p, K[i]});
20    }
21    while(!F.empty()){
22        auto [l, h] = F.top();
23        F.pop();
24        ans = max(ans, h * (n - l + 1));
25    }
26    return ans;
27 }
28 signed main(){
29     int n;
30     cin >> n;
31     for(int i = 1; i <= n; i++) cin >> K[i];
32     cout << run(n) << "\n";
33     return 0;
34 }
```

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

Map

```
1 #include <bits/stdc++.h>
2 #define int long long
3 using namespace std;
4 array<array<int, 26>, 200004> P;
5 void build(string &S){
6     set<int> M;
7     int p = 1;
8     for(char s : S){
9         for(int i = 0; i < 26; i++){
10             P[p][i] = P[p - 1][i];
11         }
12         P[p++][s - 'a']++;
13         M.insert(s - 'a');
14     }
15     for(int i = 1; i <= S.size(); i++){
16         for(int j = 25; j >= 0; j--){
17             if(M.find(j) == M.end()) continue;
18             P[i][j] -= P[i][0];
19         }
20     }
21 }
22 int run(string &S){
23     int ans = 0;
24     map<array<int, 26>, int> M;
25     for(int i = 0; i <= S.size(); i++){
26         ans += M[P[i]];
27         M[P[i]]++;
28     }
29     return ans;
30 }
31 signed main(){
32     string S;
33     cin >> S;
34     build(S);
35     cout << run(S);
36     return 0;
37 }
```

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

DP

```
1 #include <bits/stdc++.h>
2 #define int long long
3 using namespace std;
4 const int mod = 1e9 + 7;
5 array<array<int, 140004>, 504> dp;
6 int DP(int n, int k){
7     int p, sum;
8     dp[1][0] = 1;
9     for(int i = 2; i <= n; i++){
10         sum = p = 0;
11         for(int j = 0; j <= k; j++){
12             if(j - p >= i){
13                 sum -= dp[i - 1][p];
14                 p++;
15             }
16             sum += dp[i - 1][j];
17             dp[i][j] = sum % mod;
18         }
19     }
20     return dp[n][k];
21 }
22 signed main(){
23     int n, k;
24     cin >> n >> k;
25     cout << DP(n, k) << "\n";
26     return 0;
27 }
```

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

Trie

```
1 #include <bits/stdc++.h>
2 using namespace std;
3 int cnt = 0;
4 array<int, 200004> X;
5 array<array<int, 2>, 6000004> trie;
6 void update(int p, int x, int d){
7     if(d < 0) return;
8     int c = (x >> d) & 1;
9     if(!trie[p][c]) trie[p][c] = ++cnt;
10    update(trie[p][c], x, d - 1);
11 }
12 int query(int p, int x, int d){
13     if(d < 0) return x;
14     int c = ((x >> d) & 1) ^ 1;
15     if(!trie[p][c]) c ^= 1;
16     return query(trie[p][c], x, d - 1) ^ (c << d);
17 }
18 int run(int n){
19     int x = 0, ans = 0;
20     update(0, 0, 30);
21     for(int i = 1; i <= n; i++){
22         x ^= X[i];
23         ans = max(ans, query(0, x, 30));
24         update(0, x, 30);
25     }
26     return ans;
27 }
28 signed main(){
29     int n;
30     cin >> n;
31     for(int i = 1; i <= n; i++) cin >> X[i];
32     cout << run(n) << "\n";
33     return 0;
34 }
```

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

Doubling

```
1 #include <bits/stdc++.h>
2 using namespace std;
3 array<array<int, 24>, 1000004> W;
4 void build(int n){
5     for(int i = 1; i <= n; i++){
6         W[i][0] = max(W[i][0], W[i - 1][0]);
7     }
8     for(int j = 1; j < 20; j++){
9         for(int i = 1; i <= n; i++){
10            W[i][j] = W[W[i][j - 1]][j - 1];
11        }
12    }
13 }
14 int query(int l, int r){
15     int ans = 0;
16     for(int i = 19; i >= 0; i--){
17         if(W[r][i] >= l){
18             ans += 1 << i;
19             r = W[r][i];
20         }
21     }
22     return ans;
23 }
24 signed main(){
25     int n, q, l, r;
26     cin >> n >> q;
27     while(n--){
28         cin >> l >> r;
29         W[r][0] = max(l, W[r][0]);
30     }
31     build(1e6);
32     while(q--){
33         cin >> l >> r;
34         cout << query(l, r) << "\n";
35     }
36     return 0;
37 }
```

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

Greedy

```
1 #include <bits/stdc++.h>
2 #define pb push_back
3 using namespace std;
4 struct ver{
5     int u, d;
6 };
7 struct cmp{
8     bool operator()(ver a, ver b){
9         return a.d < b.d;
10    }
11 };
12 array<int, 100004> deg;
13 array<vector<int>, 100004> G;
14 bool run(int n){
15     vector<ver> tmp;
16     priority_queue<ver, vector<ver>, cmp> Q;
17     for(int i = 1; i <= n; i++){
18         Q.push({i, deg[i]});
19     }
20     while(Q.top().d > 0){
21         auto[u, d] = Q.top();
22         Q.pop();
23         while(d--){
24             auto [v, t] = Q.top();
25             Q.pop();
26             if(!t) return 0;
27             tmp.pb({v, t - 1});
28             G[u].pb(v);
29         }
30         for(ver v : tmp) Q.push(v);
31         tmp.clear();
32         Q.push({u, 0});
33     }
34     return 1;
35 }
36 signed main(){
37     int n, e = 0;
38     cin >> n;
39     for(int i = 1; i <= n; i++){
40         cin >> deg[i];
41         e += deg[i];
42     }
43     if(!run(n)){
44         cout << "IMPOSSIBLE\n";
45         return 0;
46     }
47     cout << e / 2 << "\n";
48     for(int i = 1; i <= n; i++){
49         for(int v : G[i]) cout << i << " " << v << "\n";
50     }
51     return 0;
52 }
```

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

DFS

```
1 #include <bits/stdc++.h>
2 using namespace std;
3 int p = 1;
4 array<int, 100004> pre, P;
5 void run(int l, int r){
6     int now = pre[p], mid = P[now];
7     if(mid > l){
8         p++;
9         run(l, mid - 1);
10    }
11    if(mid < r){
12        p++;
13        run(mid + 1, r);
14    }
15    cout << now << " ";
16 }
17 signed main(){
18     int n, in;
19     cin >> n;
20     for(int i = 1; i <= n; i++) cin >> pre[i];
21     for(int i = 1; i <= n; i++){
22         cin >> in;
23         P[in] = i;
24     }
25     run(1, n);
26     cout << "\n";
27     return 0;
28 }
```

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

DFS

```
1 #include <bits/stdc++.h>
2 #define pb push_back
3 using namespace std;
4 array<vector<int>, 100004> T;
5 vector<int> leaf;
6 void dfs(int u, int pre){
7     if(T[u].size() == 1) leaf.pb(u);
8     for(int v : T[u]){
9         if(v == pre) continue;
10        dfs(v, u);
11    }
12 }
13 signed main(){
14     int n, a, b;
15     cin >> n;
16     for(int i = 0; i < n - 1; i++){
17         cin >> a >> b;
18         T[a].pb(b);
19         T[b].pb(a);
20     }
21     dfs(1, 0);
22     if(leaf.size() & 1){
23         leaf.pb(leaf[0]);
24     }
25     cout << leaf.size() / 2 << "\n";
26     for(int i = 0; i < leaf.size() / 2; i++){
27         cout << leaf[i] << " " << leaf[i + leaf.size() / 2] << "\n"
28     }
29     return 0;
30 }
```

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

BFS

```
1 #include <bits/stdc++.h>
2 #define pb push_back
3 using namespace std;
4 array<int, 2504> dis;
5 array<vector<int>, 2504> G;
6 int BFS(int s){
7     int ans = 1e9;
8     queue<pair<int, int>> Q;
9     dis[s] = 1;
10    Q.push({s, 0});
11    while(!Q.empty()){
12        auto [u, pre] = Q.front();
13        Q.pop();
14        for(int v : G[u]){
15            if(v == pre) continue;
16            if(dis[v]){
17                ans = min(ans, dis[u] + dis[v] - 1);
18            }else{
19                dis[v] = dis[u] + 1;
20                Q.push({v, u});
21            }
22        }
23    }
24    return ans;
25 }
26 signed main(){
27     int n, m, a, b, ans = 1e9;
28     cin >> n >> m;
29     while(m--){
30         cin >> a >> b;
31         G[a].pb(b);
32         G[b].pb(a);
33     }
34     for(int i = 1; i <= n; i++){
35         for(int &d : dis) d = 0;
36         ans = min(ans, BFS(i));
37     }
38     if(ans >= 1e9) cout << "-1\n";
39     else cout << ans << "\n";
40     return 0;
41 }
```

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

BIT Sweeping Line

```
1 #include <bits/stdc++.h>
2 #define int long long
3 #define pb push_back
4 using namespace std;
5 struct line{
6     int p, l, r;
7 };
8 const int inf = 1e6 + 1;
9 array<int, 2000004> BIT;
10 vector<line> A, Q;
11 bool cmp(line a, line b){
12     return a.p < b.p;
13 }
14 void update(int p, int x){
15     for(; p < 2000004; p += p & -p) BIT[p] += x;
16 }
17 int query(int p){
18     int sum = 0;
19     for(; p; p -= p & -p) sum += BIT[p];
20     return sum;
21 }
22 int run(){
23     int ans = 0, p = 0;
24     for(auto [t, l, r] : Q){
25         while(p < A.size()){
26             auto [x, y, v] = A[p];
27             if(x > t) break;
28             update(y, v);
29             p++;
30         }
31         ans += query(r) - query(l - 1);
32     }
33     return ans;
34 }
35 signed main(){
36     int n, x1, x2, y1, y2;
37     cin >> n;
38     for(int i = 0; i < n; i++){
39         cin >> x1 >> y1 >> x2 >> y2;
40         x1 += inf, x2 += inf, y1 += inf, y2 += inf;
41         if(x1 == x2) Q.pb({x1, y1, y2});
42         else A.pb({x1, y1, 1}), A.pb({x2 + 1, y2, -1});
43     }
44     sort(Q.begin(), Q.end(), cmp);
45     sort(A.begin(), A.end(), cmp);
46     cout << run() << "\n";
47     return 0;
48 }
```

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

Greedy

```
1 #include <bits/stdc++.h>
2 #define int long long
3 using namespace std;
4 array<int, 1000004> P;
5 void run(int n, int k){
6     int l = 1, r = n;
7     for(int i = 1; i <= n; i++){
8         if(k >= n - i){
9             P[i] = r--;
10            k -= n - i;
11        }
12        else P[i] = l++;
13    }
14 }
15 signed main(){
16     int n, k;
17     cin >> n >> k;
18     run(n, k);
19     for(int i = 1; i <= n; i++) cout << P[i] << " ";
20     cout << "\n";
21     return 0;
22 }
```

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

分塊

```
1 #include <bits/stdc++.h>
2 using namespace std;
3
4 signed main(){
5     int t, n, k, l;
6     cin >> t;
7     while(t--){
8         l = 0;
9         cin >> n >> k;
10        if(k * k < n){
11            cout << "IMPOSSIBLE\n";
12            continue;
13        }
14        for(int i = k; i < n + k; i += k){
15            for(int j = min(i, n); j > l; j--){
16                cout << j << " ";
17            }
18            l = i;
19        }
20        cout << "\n";
21    }
22    return 0;
23 }
```

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

Greedy


```
1 #include <bits/stdc++.h>
2 using namespace std;
3 array<int, 128> C;
4 set<char> A;
5 string run(string &S){
6     int cnt = 0, n = S.size();
7     char pre = '#', mos, now;
8     string ans;
9     for(char c = 'A'; c <= 'Z'; c++){
10         if(C[c] > cnt){
11             cnt = C[c];
12             mos = c;
13         }
14     }
15     while(2 * cnt <= n){
16         cnt = 0;
17         if(pre == *A.begin()) now = *++A.begin();
18         else now = *A.begin();
19         C[now]--;
20         ans += now;
21         if(!C[now]) A.erase(now);
22         for(char c = 'A'; c <= 'Z'; c++){
23             if(C[c] > cnt){
24                 cnt = C[c];
25                 mos = c;
26             }
27         }
28         pre = now;
29         n--;
30     }
31     while(C[mos] > 1){
32         ans += mos;
33         if(mos == *A.begin()) now = *++A.begin();
34         else now = *A.begin();
35         ans += now;
36         C[now]--;
37         C[mos]--;
38         if(!C[now]) A.erase(now);
39     }
40     ans += mos;
41     return ans;
42 }
43 signed main(){
44     int n, cnt = 0;
45     string S;
46     cin >> S;
47     n = S.size();
48     for(char s : S){
49         C[s]++;
50         cnt = max(cnt, C[s]);
51         A.insert(s);
52     }
53     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

```

1 #include <bits/stdc++.h>
2 #define mid ((l + r) >> 1)
3 #define lc (p << 1)
4 #define rc ((p << 1) | 1)
5 using namespace std;
6 array<int, 800004> mix, man, tag;
7 void pull(int p){
8     mix[p] = max(mix[lc], mix[rc]);
9     man[p] = min(man[lc], man[rc]);
10 }
11 void push(int p){
12     mix[lc] += tag[p];
13     mix[rc] += tag[p];
14     man[lc] += tag[p];
15     man[rc] += tag[p];
16     tag[lc] += tag[p];
17     tag[rc] += tag[p];
18     tag[p] = 0;
19 }
20 void update(int p, int l, int r, int ql, int qr, int x){
21     if(ql > r || qr < l) return;
22     if(ql <= l && qr >= r){
23         mix[p] += x;
24         man[p] += x;
25         tag[p] += x;
26         return;
27     }
28     push(p);
29     update(lc, l, mid, ql, qr, x);
30     update(rc, mid + 1, r, ql, qr, x);
31     pull(p);
32 }
33 signed main(){
34     int n, c, s;
35     cin >> n;
36     for(int i = 0; i < n; i++){
37         cin >> c >> s;
38         update(1, 1, n, 1, c, 3 - 2 * s);
39         if(mix[1] >= 0 && man[1] >= 0) cout << ">\n";
40         else if(mix[1] <= 0 && man[1] <= 0) cout << "<\n";
41         else cout << "?\n";
42     }
43     return 0;
44 }
```

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

BIT

```
1 #include <bits/stdc++.h>
2 #define int long long
3 #define pb push_back
4 using namespace std;
5 array<int, 200004> BIT;
6 void update(int p, int x){
7     for(; p < 200004; p += p & -p) BIT[p] += x;
8 }
9 int query(int p){
10     int sum = 0;
11     for(; p; p -= p & -p) sum += BIT[p];
12     return sum;
13 }
14 signed main(){
15     int n, x, ans = 0;
16     vector<pair<int, int>> S;
17     cin >> n;
18     for(int i = 1; i <= n; i++){
19         cin >> x;
20         S.pb({x, i});
21         update(i, 1);
22     }
23     sort(S.begin(), S.end());
24     for(auto [v, p] : S){
25         ans += min(query(p - 1), query(200000) - query(p));
26         update(p, -1);
27     }
28     cout << ans << "\n";
29     return 0;
30 }
```

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

BIT DP

```
1 #include <bits/stdc++.h>
2 #define int long long
3 #define pb push_back
4 using namespace std;
5 const int mod = 1e9 + 7;
6 array<int, 200004> X, BIT;
7 void update(int p, int x){
8     for(; p < 200004; p += p & -p) BIT[p] += x;
9 }
10 int query(int p){
11     int sum = 0;
12     for(; p; p -= p & -p) sum = (sum + BIT[p]) % mod;
13     return sum;
14 }
15 void umbrella(vector<pair<int, int>> &S){
16     int lst = 0, cnt = 1;
17     sort(S.begin(), S.end());
18     for(auto [x, p] : S){
19         if(x == lst) X[p] = cnt;
20         else X[p] = ++cnt;
21         lst = x;
22     }
23 }
24 int DP(int n){
25     int ans = 0, cnt;
26     update(1, 1);
27     for(int i = 1; i <= n; i++){
28         cnt = query(X[i] - 1);
29         update(X[i], cnt);
30         ans = (ans + cnt) % mod;
31     }
32     return ans;
33 }
34 signed main(){
35     int n, x;
36     vector<pair<int, int>> S;
37     cin >> n;
38     for(int i = 1; i <= n; i++){
39         cin >> x;
40         S.pb({x, i});
41     }
42     umbrella(S);
43     cout << DP(n) << "\n";
44     return 0;
45 }
```

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

DP

```
1 #include <bits/stdc++.h>
2 #define int long long
3 using namespace std;
4 int mod = 1e9 + 7;
5 array<int, 500004> dp;
6 int DP(string &S){
7     int n = S.size(), ans = 0;
8     dp[0] = 1;
9     for(int i = 1; i <= n; i++){
10         for(int j = i - 1; j >= 0; j--){
11             dp[i] = (dp[i] + dp[j]) % mod;
12             if(!j || S[i - 1] == S[j - 1]) break;
13         }
14         ans = (ans + dp[i]) % mod;
15     }
16     return ans;
17 }
18 signed main(){
19     string S;
20     cin >> S;
21     cout << DP(S) << "\n";
22     return 0;
23 }
```

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

Segment Tree

```
1 #include <bits/stdc++.h>
2 #define mid ((l + r) >> 1)
3 #define lc (p << 1)
4 #define rc ((p << 1) | 1)
5 using namespace std;
6 array<array<int, 800004>, 2> seg, pre, suf, all;
7 void pull(int p, int l, int r){
8     for(int i : {0, 1}){
9         all[i][p] = all[i][lc] & all[i][rc];
10        if(all[i][p]){
11            pre[i][p] = suf[i][p] = seg[i][p] = r - l + 1;
12            continue;
13        }
14        if(all[i][lc]) pre[i][p] = (mid - l + 1) + pre[i][rc];
15        else pre[i][p] = pre[i][lc];
16        if(all[i][rc]) suf[i][p] = (r - mid) + suf[i][lc];
17        else suf[i][p] = suf[i][rc];
18        seg[i][p] = max({seg[i][lc], seg[i][rc], pre[i][p], suf[i][p]});
19    }
20 }
21 void update(int p, int l, int r, int c, int x){
22     if(c > r || c < l) return;
23     if(l == r){
24         seg[x][p] = pre[x][p] = suf[x][p] = all[x][p] = 1;
25         seg[x ^ 1][p] = pre[x ^ 1][p] = suf[x ^ 1][p] = all[x ^ 1][p];
26         return;
27     }
28     update(lc, l, mid, c, x);
29     update(rc, mid + 1, r, c, x);
30     pull(p, l, r);
31 }
32 signed main(){
33     int n, q, p;
34     string S;
35     cin >> S;
36     n = S.size();
37     for(int i = 1; i <= n; i++){
38         update(1, 1, n, i, S[i - 1] ^ 48);
39     }
40     cin >> q;
41     while(q--){
42         cin >> p;
43         update(1, 1, n, p, S[p - 1] ^ 49);
44         cout << max(seg[0][1], seg[1][1]) << " ";
45         S[p - 1] ^= 1;
46     }
47     cout << "\n";
48     return 0;
49 }
```

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

Lucas Theorem

```
1 #include <bits/stdc++.h>
2 using namespace std;
3 int C(int n, int k){
4     if(n == k || !k) return 1;
5     if(k > n) return 0;
6     return C(n >> 1, k >> 1) & C(n & 1, k & 1);
7 }
8 signed main(){
9     int n, x, sum = 0;
10    cin >> n;
11    for(int i = 0; i < n; i++){
12        cin >> x;
13        if(C(n - 1, i)) sum ^= x;
14    }
15    cout << sum << "\n";
16    return 0;
17 }
```

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

二分搜

```
1 #include <bits/stdc++.h>
2 #define int long long
3 using namespace std;
4 int see(int n){
5     int ans = 0, cnt;
6     for(int i = 1; i < 10; i++){
7         cnt = 0;
8         for(int t = 1; t <= n; t *= 10){
9             cnt += (n / (10 * t)) * t;
10            cnt += min(t, max(0ll, n % (10 * t) - i * t + 1));
11        }
12        ans = max(ans, cnt);
13    }
14    return ans;
15 }
16 int BIS(int k){
17     int l = 0, r = 1e18, mid;
18     while(l != r){
19         mid = ((l + r) >> 1) + 1;
20         if(see(mid) <= k) l = mid;
21         else r = mid - 1;
22     }
23     return l;
24 }
25 signed main(){
26     int n;
27     cin >> n;
28     cout << BIS(n) << "\n";
29     return 0;
30 }
```

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

爆搜

```

1 #include <bits/stdc++.h>
2 #define pb push_back
3 using namespace std;
4 array<vector<char>, 128> C;
5 array<vector<int>, 128> I;
6 array<int, 128> P;
7 string trans(string &S){
8     int p = 0;
9     char now = '#', tmp;
10    string ans;
11    for(char s : S){
12        C[s].pb(s);
13        I[s].pb(0);
14    }
15    for(int i = 0; i < 128; i++){
16        for(int j = 0; j < C[i].size(); j++){
17            C[i][j] = S[p];
18            I[i][j] = P[S[p++]]++;
19        }
20    }
21    p = 0;
22    for(int i = 1; i < S.size(); i++){
23        tmp = now;
24        now = C[now][p];
25        p = I[tmp][p];
26        ans += now;
27    }
28    reverse(ans.begin(), ans.end());
29    return ans;
30 }
31 signed main(){
32     string S;
33     cin >> S;
34     cout << trans(S) << "\n";
35     return 0;
36 }
```

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

1 噗噗待哺

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

Monotone Stack

```

1 #include <bits/stdc++.h>
2 #define ff first
3 #define ss second
4 using namespace std;
5 array<int, 1004> H;
6 array<array<char, 1004>, 1004> G;
7 stack<pair<int, int>> S;
8 int run(int n, int m){
9     int ans = 0, now;
10    for(int j = 1; j <= n; j++){
11        S.push({0, 0});
12        for(int i = 1; i <= m; i++){
13            now = i;
14            if(G[j][i] == '*') H[i] = 0;
15            else H[i]++;
16            while(!S.empty() && H[i] <= S.top().ff){
17                auto [h, p] = S.top();
18                S.pop();
19                ans = max(ans, (i - p) * h);
20                now = p;
21            }
22            S.push({H[i], now});
23        }
24        while(!S.empty()){
25            auto [h, p] = S.top();
26            S.pop();
27            ans = max(ans, (m - p + 1) * h);
28        }
29    }
30    return ans;
31 }
32 signed main(){
33     int n, m;
34     cin >> n >> m;
35     for(int i = 1; i <= n; i++){
36         for(int j = 1; j <= m; j++){
37             cin >> G[i][j];
38         }
39     }
40     cout << run(n, m) << "\n";
41     return 0;
42 }
```

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

BIT LIS DFS

```
1 #include <bits/stdc++.h>
2 #define int long long
3 using namespace std;
4 bitset<200004> vis;
5 array<int, 200004> P, pre, mix;
6 void update(int t, int p, int x){
7     for(; p < 200004; p += p & -p){
8         if(t) mix[p] = max(mix[p], x);
9         else pre[p] += x;
10    }
11 }
12 int query(int t, int p){
13     int ans = 0;
14     for(; p; p -= p & -p){
15         if(t) ans = max(ans, mix[p]);
16         else ans += pre[p];
17     }
18     return ans;
19 }
20 void DFS(int u){
21     if(vis[u]) return;
22     vis[u] = 1;
23     DFS(P[u]);
24 }
25 signed main(){
26     int n, lis = 0, inv = 0, cyc = 0, dec = 0;
27     cin >> n;
28     for(int i = 1; i <= n; i++){
29         cin >> P[i];
30         update(0, P[i], 1);
31         inv += query(0, 200000) - query(0, P[i]);
32         lis = max(lis, query(1, P[i]) + 1);
33         update(1, P[i], query(1, P[i]) + 1);
34     }
35     for(int i = n, p = n; i; i--){
36         if(!vis[i]){
37             DFS(i);
38             cyc++;
39         }
40         if(P[i] == p){
41             p--;
42             dec++;
43         }
44     }
45     cout << inv << " " << n - cyc << " " << n - lis << " " << n - d
46     return 0;
47 }
```

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

三分搜

```
1 #include <bits/stdc++.h>
2 #define int long long
3 using namespace std;
4 int n, k;
5 array<int, 200004> X;
6 int see(int p){
7     int sum = 0, cnt = 0, s, t;
8     for(s = 1; s <= n; s++){
9         if(sum + X[s] > p) break;
10        sum += X[s];
11    }
12    for(t = n; t >= s; t--){
13        if(sum + X[t] > k) break;
14        sum += X[t];
15    }
16    if(sum) cnt++;
17    sum = 0;
18    for(int i = s; i <= t; i++){
19        if(sum + X[i] > k){
20            sum = 0;
21            cnt++;
22        }
23        sum += X[i];
24    }
25    if(sum) cnt++;
26    return cnt;
27 }
28 int TIS(){
29     int l = 0, r = k, lm, rm, ans = 1e9;
30     while(r - l > 2){
31         lm = (2 * l + r) / 3ll;
32         rm = (l + 2 * r) / 3ll;
33         if(see(lm) < see(rm)) r = rm;
34         else l = lm;
35     }
36     for(int i = l; i <= r; i++){
37         ans = min(ans, see(i));
38     }
39     return ans;
40 }
41 signed main(){
42     cin >> n >> k;
43     for(int i = 1; i <= n; i++){
44         cin >> X[i];
45     }
46     cout << TIS() << "\n";
47     return 0;
48 }
```

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

爆搜

```
1 #include <bits/stdc++.h>
2 #define pb push_back
3 using namespace std;
4 array<int, 104> A;
5 vector<int> B;
6 bool gen(int n, int a1a2){
7     int a0ai;
8     multiset<int> BB;
9     for(int b : B) BB.insert(b);
10    A[0] = (B[0] + B[1] - a1a2) / 2;
11    A[1] = B[0] - A[0];
12    A[2] = B[1] - A[0];
13    BB.erase(BB.find(B[0]));
14    BB.erase(BB.find(B[1]));
15    BB.erase(BB.find(a1a2));
16    for(int i = 3; i < n; i++){
17        a0ai = *BB.begin();
18        A[i] = a0ai - A[0];
19        for(int j = 0; j < i; j++){
20            if(BB.find(A[i] + A[j]) == BB.end()) return 0;
21            BB.erase(BB.find(A[i] + A[j]));
22        }
23    }
24    for(int i = 1; i < n; i++){
25        if(A[i] < A[i - 1]) return 0;
26    }
27    return 1;
28 }
29 signed main(){
30     int n, b;
31     cin >> n;
32     for(int i = 0; i < n * (n - 1) / 2; i++){
33         cin >> b;
34         B.pb(b);
35     }
36     sort(B.begin(), B.end());
37     for(int i = 2; i <= n; i++){
38         if(gen(n, B[i])){
39             for(int j = 0; j < n; j++) cout << A[j] << " ";
40             break;
41         }
42     }
43     cout << "\n";
44     return 0;
45 }
```

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

DP圖

```
1 #include <bits/stdc++.h>
2 #define int long long
3 using namespace std;
4
5 signed main(){
6     int n, x, ans = 0;
7     priority_queue<int> Q;
8     cin >> n;
9     for(int i = 0; i < n; i++){
10         cin >> x;
11         Q.push(x);
12         ans += Q.top() - x;
13         Q.pop();
14         Q.push(x);
15     }
16     cout << ans << "\n";
17     return 0;
18 }
```

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

三分搜

```
1 #include <bits/stdc++.h>
2 #define int long long
3 using namespace std;
4 array<int, 200004> A, B, cur;
5 int run(int n, int p){
6     int cnt = llabs(p);
7     for(int i = 1; i <= n; i++){
8         cur[i] = A[i];
9     }
10    cur[n] -= p;
11    cur[1] += p;
12    for(int i = 1; i < n; i++){
13        cnt += llabs(cur[i] - B[i]);
14        cur[i + 1] += cur[i] - B[i];
15        cur[i] = B[i];
16    }
17    return cnt;
18 }
19 int TIS(int n){
20     int l = -1e12, r = 1e12, lmid, rmid;
21     while(r - l > 2){
22         lmid = (2 * l + r) / 3;
23         rmid = (l + 2 * r) / 3;
24         if(run(n, lmid) < run(n, rmid)) r = rmid;
25         else l = lmid;
26     }
27     return min({run(n, l), run(n, l + 1), run(n, r)});
28 }
29 signed main(){
30     int n;
31     cin >> n;
32     for(int i = 1; i <= n; i++) cin >> A[i];
33     for(int i = 1; i <= n; i++) cin >> B[i];
34     cout << TIS(n) << "\n";
35     return 0;
36 }
```

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

SOS DP

```

1 #include <bits/stdc++.h>
2 using namespace std;
3 const int C = (1 << 20) - 1;
4 array<int, 200004> X, O, A, R;
5 array<int, 1 << 20> dp;
6 void OR(int n){
7     for(int &d : dp) d = 0;
8     for(int i = 1; i <= n; i++) dp[X[i]]++;
9     for(int k = 1; k < 1 << 20; k <= 1){
10         for(int i = 0; i < 1 << 20; i++){
11             if(i & k) dp[i] += dp[i ^ k];
12         }
13     }
14     for(int i = 1; i <= n; i++){
15         O[i] = dp[X[i]];
16         R[i] = n - dp[C ^ X[i]];
17     }
18 }
19 void AND(int n){
20     for(int &d : dp) d = 0;
21     for(int i = 1; i <= n; i++) dp[X[i]]++;
22     for(int k = 1 << 19; k; k >>= 1){
23         for(int i = 0; i < 1 << 20; i++){
24             if(i & k) continue;
25             dp[i] += dp[i ^ k];
26         }
27     }
28     for(int i = 1; i <= n; i++){
29         A[i] = dp[X[i]];
30     }
31 }
32 signed main(){
33     int n;
34     cin >> n;
35     for(int i = 1; i <= n; i++) cin >> X[i];
36     OR(n);
37     AND(n);
38     for(int i = 1; i <= n; i++){
39         cout << O[i] << " " << A[i] << " " << R[i] << "\n";
40     }
41     return 0;
42 }
```

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

通靈


```
1 #include <bits/stdc++.h>
2 #define pb push_back
3 using namespace std;
4 int cnt = 0;
5 bitset<200004> vis;
6 array<int, 200004> G, R;
7 array<vector<pair<int, int>>, 200004> ans;
8 void cut(int u){
9     int l, r, nl, nr;
10    if(G[u] == u || vis[u]) return;
11    if(R[u] == G[u]){
12        if(vis[G[u]]) return;
13        ans[cnt].pb({u, G[u]});
14        vis[u] = vis[G[u]] = 1;
15        swap(G[u], G[G[u]]);
16        swap(R[u], R[R[u]]);
17    }else{
18        while(1){
19            l = R[u], r = G[u];
20            nl = R[l], nr = G[r];
21            ans[cnt].pb({l, r});
22            vis[l] = vis[r] = 1;
23            if(nr == l && nl == r){
24                R[l] = l;
25                R[u] = r;
26                swap(G[l], G[r]);
27                return;
28            }else{
29                R[u] = r;
30                R[nr] = l;
31                swap(G[l], G[r]);
32                u = l;
33            }
34            if(nl == nr) return;
35            u = l;
36        }
37    }
38 }
39 void run(int n){
40     bool ok;
41     while(1){
42         ok = 1;
43         for(int i = 1; i <= n; i++){
44             vis[i] = 0;
45             ok &= G[i] == i;
46         }
47         if(ok) return;
48         for(int i = 1; i <= n; i++) cut(i);
49         cnt++;
50     }
51 }
52 signed main(){
53     int n, x;
```

```
54     cin >> n;
55     for(int i = 1; i <= n; i++){
56         cin >> x;
57         G[i] = x;
58         R[x] = i;
59     }
60     run(n);
61     cout << cnt << "\n";
62     for(int i = 0; i < cnt; i++){
63         cout << ans[i].size() << "\n";
64         for(auto [u, v] : ans[i]){
65             cout << u << " " << v << "\n";
66         }
67     }
68     return 0;
69 }
```

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

逆DP

```
1 #include <bits/stdc++.h>
2 using namespace std;
3 const int inf = 1 << 20;
4 string ans;
5 int RUN(int a, int b){
6     if(!a && !b) return 0;
7     if(a == b) return inf;
8     return RUN(a % (b + 1), b % (a + 1)) + a / (b + 1) + b / (a + 1)
9 }
10 void back(int a, int b){
11     if(!a && !b) return;
12     if(a > b){
13         back(a - b - 1, b);
14         ans += '0';
15     }
16     else{
17         back(a, b - a - 1);
18         ans += '1';
19     }
20 }
21 signed main(){
22     int n, cnt = inf, tmp, a, b;
23     cin >> n;
24     for(int i = 0; i <= n; i++){
25         tmp = RUN(i, n - i);
26         if(tmp < cnt){
27             a = i, b = n - i;
28             cnt = tmp;
29         }
30     }
31     back(a, b);
32     cout << ans << "\n";
33     return 0;
34 }
```

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

Hash


```
1 #include <bits/stdc++.h>
2 #define int long long
3 #define pb push_back
4 using namespace std;
5 const int mod = 1000696969, c = 41;
6 array<int, 100004> C, S1, S2;
7 array<vector<int>, 100004> T1, T2;
8 void build(int n){
9     C[0] = 1;
10    for(int i = 1; i <= n; i++){
11        C[i] = c * C[i - 1] % mod;
12    }
13 }
14 int DFS(array<vector<int>, 100004> &T, array<int, 100004> &S, int u,
15         int sum = 0;
16         vector<pair<int, int>> H;
17         for(int v : T[u]){
18             if(v == pre) continue;
19             H.pb({DFS(T, S, v, u), S[v]});
20         }
21         sort(H.begin(), H.end());
22         for(auto [h, s] : H){
23             sum = (sum + h * C[S[u]]) % mod;
24             S[u] += s;
25         }
26         S[u]++;
27         sum = (sum + S[u] * C[S[u]]) % mod;
28         return sum;
29     }
30 signed main(){
31     int t, n, a, b;
32     build(100000);
33     cin >> t;
34     while(t--){
35         cin >> n;
36         for(int i = 1; i <= n; i++){
37             S1[i] = S2[i] = 0;
38             T1[i].clear();
39             T2[i].clear();
40         }
41         for(int i = 1; i < n; i++){
42             cin >> a >> b;
43             T1[a].pb(b);
44             T1[b].pb(a);
45         }
46         for(int i = 1; i < n; i++){
47             cin >> a >> b;
48             T2[a].pb(b);
49             T2[b].pb(a);
50         }
51         if(DFS(T1, S1, 1, 0) == DFS(T2, S2, 1, 0)) cout << "YES\n";
52         else cout << "NO\n";
53     }
```

```
54     return 0;
55 }
```

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

Math

```
1 #include <bits/stdc++.h>
2 #define int long long
3 using namespace std;
4 const int mod = 1e9 + 7;
5 array<int, 1000004> fac;
6 void build(int n){
7     fac[0] = 1;
8     for(int i = 1; i <= n; i++){
9         fac[i] = i * fac[i - 1] % mod;
10    }
11 }
12 int exp(int x, int k){
13     int ans = 1;
14     for(int i = 1; i <= k; i <= 1){
15         if(i & k) ans = ans * x % mod;
16         x = x * x % mod;
17     }
18     return ans;
19 }
20 int C(int n, int k){
21     return (fac[n] * exp(fac[n - k], mod - 2) % mod) * exp(fac[k],
22 }
23 int math(int n, int k){
24     int ans = 0;
25     for(int i = 0; i <= k; i++){
26         ans = (ans + ((exp(-1, k - i) * C(k, i) % mod) * exp(i, n)
27     }
28     return ans;
29 }
30 signed main(){
31     int n, k;
32     cin >> n >> k;
33     build(1000000);
34     cout << math(n, k) << "\n";
35     return 0;
36 }
```

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

Dominator Tree


```
1 #include <bits/stdc++.h>
2 #define pb push_back
3 using namespace std;
4 int cnt = 0;
5 array<int, 100004> P, rev, L, DSU, pre, sdom, idom;
6 array<vector<int>, 100004> G, R, buk;
7 vector<int> ans;
8 int find(int u, int x = 0){
9     if(DSU[u] == u) return x? -1 : u;
10    int v = find(DSU[u], x + 1);
11    if(v < 0) return u;
12    if(sdom[L[DSU[u]]] < sdom[L[u]]) L[u] = L[DSU[u]];
13    DSU[u] = v;
14    return x? v : L[u];
15 }
16 void onion(int u, int v){
17     DSU[v] = u;
18 }
19 void DFS(int u){
20     P[u] = ++cnt, rev[cnt] = u, L[cnt] = cnt, DSU[cnt] = cnt, sdom[cnt] = 0;
21     for(int v : G[u]){
22         if(!P[v]){
23             DFS(v);
24             pre[P[v]] = P[u];
25         }
26         R[P[v]].pb(P[u]);
27     }
28 }
29 void run(int u){
30     ans.pb(rev[u]);
31     if(u == 1) return;
32     run(idom[u]);
33 }
34 signed main(){
35     int n, m, a, b, w;
36     cin >> n >> m;
37     while(m--){
38         cin >> a >> b;
39         G[a].pb(b);
40     }
41     DFS(1);
42     for(int u = n; u; u--){
43         for(int v : R[u]){
44             sdom[u] = min(sdom[u], sdom[find(v)]);
45         }
46         if(u > 1) buk[sdom[u]].pb(u);
47         for(int v : buk[u]){
48             w = find(v);
49             if(sdom[v] == sdom[w]) idom[v] = sdom[v];
50             else idom[v] = w;
51         }
52         onion(pre[u], u);
53     }
```

```
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

```
1 #include <bits/stdc++.h>
2 using namespace std;
3 int p = 0;
4 array<int, 100004> DSU, S, C;
5 array<array<int, 100004>, 2> dp;
6 int find(int u){
7     if(DSU[u] == u) return u;
8     return DSU[u] = find(DSU[u]);
9 }
10 void onion(int u, int v){
11     int U = find(u), V = find(v);
12     if(U == V) return;
13     if(S[U] < S[V]) swap(U, V);
14     DSU[V] = U;
15     S[U] += S[V];
16 }
17 void DP(int n){
18     int sum;
19     dp[0][0] = 1;
20     for(int i = 1; i <= n; i++){
21         if(!C[i]) continue;
22         p ^= 1;
23         for(int j = 0; j < i; j++){
24             sum = 0;
25             for(int k = j; k <= n; k += i){
26                 if(k - j > i * C[i]) sum -= dp[p ^ 1][k - i * (C[i]
27                                         sum += dp[p ^ 1][k];
28                                         dp[p][k] = sum? 1 : 0;
29             }
30         }
31     }
32 }
33 signed main(){
34     int n, m, a, b;
35     cin >> n >> m;
36     for(int i = 1; i <= n; i++){
37         DSU[i] = i;
38         S[i] = 1;
39     }
40     while(m--){
41         cin >> a >> b;
42         onion(a, b);
43     }
44     for(int i = 1; i <= n; i++){
45         if(DSU[i] != i) continue;
46         C[S[i]]++;
47     }
48     DP(n);
49     for(int i = 1; i <= n; i++) cout << dp[p][i];
50     cout << "\n";
51     return 0;
52 }
```

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

Dinic


```
1 #include <bits/stdc++.h>
2 #define pb push_back
3 using namespace std;
4 struct edge{
5     int u, v, f;
6 };
7 int inf = 1000696969, cnt = 0;
8 bitset<204> vis;
9 array<int, 204> lvl, P;
10 array<edge, 84004> E;
11 array<vector<int>, 204> G;
12 void add(int u, int v, int f){
13     E[cnt] = {u, v, 1};
14     G[u].pb(cnt++);
15     E[cnt] = {v, u, 0};
16     G[v].pb(cnt++);
17 }
18 bool BFS(int s, int t){
19     int u;
20     queue<int> Q;
21     Q.push(s);
22     lvl[s] = 1;
23     while(!Q.empty()){
24         u = Q.front();
25         Q.pop();
26         for(int i : G[u]){
27             auto [o, v, f] = E[i];
28             if(!f || lvl[v]) continue;
29             lvl[v] = lvl[u] + 1;
30             Q.push(v);
31         }
32     }
33     return lvl[t];
34 }
35 int DFS(int u, int t, int f){
36     if(u == t || !f) return f;
37     int wat = 0, tmp;
38     for(int &i = P[u]; i < G[u].size(); i++){
39         auto &eu, ev, ef = E[G[u][i]];
40         auto &bu, bv, bf = E[G[u][i] ^ 1];
41         if(lvl[ev] != lvl[u] + 1) continue;
42         tmp = DFS(ev, t, min(f, ef));
43         ef -= tmp, bf += tmp;
44         f -= tmp, wat += tmp;
45     }
46     return wat;
47 }
48 int flow(int s, int t){
49     int tmp, ans = 0;
50     while(1){
51         for(int &l : lvl) l = 0;
52         if(!BFS(s, t)) break;
53         while(1){
```

```
54         for(int &p : P) p = 0;
55         if(tmp = DFS(s, t, inf)) ans += tmp;
56         else break;
57     }
58 }
59 return ans;
60 }
61 void run(int u){
62     if(vis[u]) return;
63     vis[u] = 1;
64     for(int i : G[u]){
65         auto [o, v, f] = E[i];
66         if(!f) continue;
67         run(v);
68     }
69 }
70 signed main(){
71     int n;
72     char c;
73     cin >> n;
74     for(int i = 1; i <= n; i++){
75         add(0, i, 1);
76         add(100 + i, 201, 1);
77     }
78     for(int i = 1; i <= n; i++){
79         for(int j = 1; j <= n; j++){
80             cin >> c;
81             if(c == '.') continue;
82             add(i, 100 + j, 1);
83         }
84     }
85     cout << flow(0, 201) << "\n";
86     run(0);
87     for(int i = 1; i <= n; i++){
88         if(!vis[i]) cout << "1 " << i << "\n";
89         if(vis[100 + i]) cout << "2 " << i << "\n";
90     }
91     return 0;
92 }
```

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

Binary Search Segment Tree


```
1 #include <bits/stdc++.h>
2 #define int long long
3 #define pb push_back
4 using namespace std;
5 struct line{
6     int t, p, l, r;
7 };
8 int N = 1 << 18;
9 array<char, 100004> F;
10 array<int, 1 << 19> seg, tag, D;
11 array<vector<pair<int, int>>, 200004> X, Y;
12 vector<int> P;
13 vector<line> S, Q;
14 map<int, int> O;
15 bool cmp(line a, line b){
16     if(a.p == b.p) return abs(a.t) > abs(b.t);
17     return a.p < b.p;
18 }
19 void update(int l, int r, int x){
20     int cntl = 0, cntr = 0;
21     l += N - 1, r += N + 1;
22     for(int n = 1; (l ^ r) > 1; l >= 1, r >= 1, n <= 1){
23         seg[l] += cntl * x, seg[r] += cntr * x;
24         if(~l & 1) tag[l ^ 1] += x, seg[l ^ 1] += n * x, cntl += n;
25         if(r & 1) tag[r ^ 1] += x, seg[r ^ 1] += n * x, cntr += n;
26     }
27     for(; l; l >= 1, r >= 1) seg[l] += cntl * x, seg[r] += cntr;
28 }
29 int query(int l, int r){
30     int sum = 0, cntl = 0, cntr = 0;
31     l += N - 1, r += N + 1;
32     for(int n = 1; (l ^ r) > 1; l >= 1, r >= 1, n <= 1){
33         sum += tag[l] * cntl + tag[r] * cntr;
34         if(~l & 1) sum += seg[l ^ 1], cntl += n;
35         if(r & 1) sum += seg[r ^ 1], cntr += n;
36     }
37     for(; r; r >= 1) sum += cntl * tag[l] + cntr * tag[r];
38     return sum;
39 }
40 bool banana(int p){
41     int ans = 0;
42     Q.clear();
43     for(int i = 0; i < p; i++){
44         auto [x1, x2, y1, y2] = S[i];
45         if(x1 == x2){
46             if(y1 > y2) swap(y1, y2);
47             Q.pb({0, x1, y1, y2});
48             X[x2].pb({y1, y2});
49         }else{
50             if(x1 > x2) swap(x1, x2);
51             Q.pb({1, x1, y1, 0});
52             Q.pb({-1, x2 + 1, y2, 0});
53             Y[y1].pb({x1, x2});
54         }
55     }
56 }
```

```
54         }
55     }
56     sort(Q.begin(), Q.end(), cmp);
57     for(auto [t, p, l, r] : Q){
58         if(t) update(l, l, t);
59         else ans |= query(l, r);
60     }
61     for(int i = 1; i < 200004; i++){
62         for(auto [l, r] : X[i]){
63             ans |= query(l, r);
64             update(l, r, 1);
65         }
66         for(auto [l, r] : X[i]){
67             update(l, r, -1);
68         }
69         for(auto [l, r] : Y[i]){
70             ans |= query(l, r);
71             update(l, r, 1);
72         }
73         for(auto [l, r] : Y[i]){
74             update(l, r, -1);
75         }
76         X[i].clear(), Y[i].clear();
77     }
78     return ans;
79 }
80 int crash(int n){
81     if(F[n] == 'U' && F[n - 1] == 'D' || F[n] == 'D' && F[n - 1] =
82     if(F[n] == 'L' && F[n - 1] == 'R' || F[n] == 'R' && F[n - 1] =
83     auto [sx, tx, sy, ty] = S[n];
84     for(int i = 0; i < n; i++){
85         auto [x1, x2, y1, y2] = S[i];
86         if(y1 > y2) swap(y1, y2);
87         if(x1 > x2) swap(x1, x2);
88         if(sy == ty){
89             if(y1 <= sy && y2 >= ty) update(x1, x2, 1);
90         }else{
91             if(x1 <= sx && x2 >= tx) update(y1, y2, 1);
92         }
93     }
94     if((sy == ty && sx <= tx) || (sx == tx && sy <= ty)){
95         for(int i = (sx == tx? sy : sx); i <= (sx == tx? ty : tx);
96             if(query(i, i)) return 0[i] - 0[(sx == tx? sy : sx)] +
97         }
98     }else{
99         for(int i = (sx == tx? sy : sx); i >= (sx == tx? ty : tx);
100            if(query(i, i)) return 0[(sx == tx? sy : sx)] - 0[i] +
101        }
102    }
103    return 1;
104 }
105 int BIS(int n){
106     int p = 0;
107 }
```

```

107     TOR(int i = 1 << 10; i; i >>= 1){  

108         if(p + i <= n && !banana(p + i)) p += i;  

109     }  

110     return p;  

111 }  

112 void crepe(vector<int> &X, vector<line> &L){  

113     int p = 0;  

114     map<int, int> M;  

115     for(int x : X) M[x] = 0;  

116     for(auto &[x, c] : M) c = ++p, 0[p] = x;  

117     for(auto &[x1, x2, y1, y2] : L){  

118         x1 = M[x1], x2 = M[x2];  

119         y1 = M[y1], y2 = M[y2];  

120     }  

121 }  

122 signed main(){  

123     cin.tie(0), cout.tie(0), ios::sync_with_stdio(0);  

124     char d;  

125     int n, t, x = 0, y = 0, px, py, p;  

126     cin >> n;  

127     P.pb(0);  

128     for(int i = 1; i <= n; i++){  

129         cin >> d >> t;  

130         D[i] = D[i - 1] + t;  

131         if(d == 'U' || d == 'D'){  

132             py = y + (d == 'U'? 1 : -1), y += (d == 'U'? t : -t);  

133             if(i == 1) py += (d == 'U'? -1 : 1);  

134             S.pb({x, x, py, y});  

135             P.pb(py), P.pb(y);  

136         }else{  

137             px = x + (d == 'R'? 1 : -1), x += (d == 'R'? t : -t);  

138             if(i == 1) px += (d == 'R'? -1 : 1);  

139             S.pb({px, x, y, y});  

140             P.pb(px), P.pb(x);  

141         }  

142         F[i - 1] = d;  

143     }  

144     crepe(P, S);  

145     p = BIS(n);  

146     if(p == n) cout << D[n] << "\n";  

147     else cout << D[p] + crash(p) << "\n";  

148     return 0;  

149 }

```

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

Greedy


```
1 #include <bits/stdc++.h>
2 #define int long long
3 #define pb push_back
4 #define pii pair<int, int>
5 #define ff first
6 #define ss second
7 using namespace std;
8 bitset<200004> pro, art, awy;
9 array<int, 200004> X, Y;
10 vector<pair<int, int>> P, A;
11 priority_queue<int> sas;
12 priority_queue<int, vector<int>, greater<int>> bas;
13 priority_queue<pii> de;
14 int chos(int a, int b){
15     int sum = 0, ans = 0;
16     for(int i = 0; i < a + b; i++){
17         auto [x, j] = P[i];
18         sum += Y[j];
19         de.push({x - Y[j], j});
20         art[j] = 1;
21     }
22     for(int i = a + b; i < P.size(); i++){
23         auto [x, j] = P[i];
24         sas.push(Y[j]);
25     }
26     sas.push(0);
27     while(de.size() > b){
28         auto [d, i] = de.top();
29         de.pop();
30         sum += d;
31         pro[i] = 1;
32         art[i] = 0;
33     }
34     ans = max(ans, sum);
35     for(int i = a + b - 1; i >= a; i--){
36         auto [x, j] = P[i];
37         awy[j] = 1;
38         if(art[j]){
39             art[j] = 0;
40             sum -= Y[j];
41         }else{
42             pro[j] = 0;
43             sum -= x;
44             while(awy[de.top().ss]) de.pop();
45             auto [d, k] = de.top();
46             de.pop();
47             sum += d;
48             pro[k] = 1;
49             art[k] = 0;
50         }
51         bas.push(Y[j]);
52         sum += Y[j];
53         if(sas.top() > bas.top()){
```

```
54         sum += sas.top() - bas.top();
55         sas.push(bas.top());
56         bas.push(sas.top());
57         bas.pop(), sas.pop();
58     }
59     ans = max(ans, sum);
60 }
61 return ans;
62 }
63 signed main(){
64     int a, b, n;
65     cin >> a >> b >> n;
66     for(int i = 1; i <= n; i++){
67         cin >> X[i] >> Y[i];
68         P.pb({X[i], i});
69         A.pb({Y[i], i});
70     }
71     sort(P.begin(), P.end(), greater<pii>());
72     sort(A.begin(), A.end(), greater<pii>());
73     cout << chos(a, b) << "\n";
74     return 0;
75 }
```

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

Topological Sort

```

1 #include <bits/stdc++.h>
2 #define pb push_back
3 using namespace std;
4 array<int, 100004> in;
5 array<vector<int>, 100004> G;
6 vector<int> ord;
7 void topu(int n){
8     int u;
9     priority_queue<int> Q;
10    for(int i = 1; i <= n; i++){
11        if(!in[i]) Q.push(i);
12    }
13    while(!Q.empty()){
14        u = Q.top();
15        Q.pop();
16        ord.pb(u);
17        for(int v : G[u]){
18            if(!--in[v]) Q.push(v);
19        }
20    }
21 }
22 signed main(){
23     int n, m, a, b;
24     cin >> n >> m;
25     while(m--){
26         cin >> a >> b;
27         in[a]++;
28         G[b].pb(a);
29     }
30     topu(n);
31     reverse(ord.begin(), ord.end());
32     for(int u : ord) cout << u << " ";
33     cout << "\n";
34     return 0;
35 }
```

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

1 啾啾待哺

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

Greedy

```
1 #include <bits/stdc++.h>
2 #define int long long
3 using namespace std;
4 array<array<int, 3>, 100004> C;
5 int RUN(int n){
6     int cnt = 0;
7     for(int i = 1; i <= n; i++){
8         if(C[i][1] * C[i][2] < 0){
9             if(abs(C[i][1]) > abs(C[i][2])){
10                 cnt += abs(C[i][2]);
11                 C[i][1] += C[i][2];
12                 C[i][2] = 0;
13             }else{
14                 cnt += abs(C[i][1]);
15                 C[i][2] += C[i][1];
16                 C[i][1] = 0;
17             }
18         }
19         cnt += abs(C[i][1] + C[i][2]);
20         C[i + 1][1] += C[i][1], C[i + 1][2] += C[i][2];
21     }
22     return cnt;
23 }
24 signed main(){
25     int n;
26     cin >> n;
27     for(int i : {1, 2}){
28         for(int j = 1; j <= n; j++){
29             cin >> C[j][i];
30             C[j][i] -= 1;
31         }
32     }
33     cout << RUN(n) << "\n";
34     return 0;
35 }
```

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

DP

```
1 #include <bits/stdc++.h>
2 #define int long long
3 using namespace std;
4 const int mod = 1e9 + 7, two = 5e8 + 4;
5 array<array<int, 250004>, 504> dp;
6 int DP(int n, int k){
7     int ans = 0;
8     dp[1][0] = 1, dp[1][1] = 2;
9     dp[2][0] = 1, dp[2][1] = 4, dp[2][2] = 2;
10    for(int i = 3; i < n; i++){
11        for(int j = 0; j <= min(i, k); j++){
12            dp[i][j] = (dp[i][j] + dp[i - 2][j]) % mod;
13            if(j) dp[i][j] = (dp[i][j] + dp[i - 2][j - 1] * 2 * (i - j)) % mod;
14            if(j >= 2) dp[i][j] = (dp[i][j] + (dp[i - 2][j - 2] * 2 * (i - j) + dp[i - 2][j - 1] * (i - j))) % mod;
15        }
16    }
17    if(n > 2){
18        for(int j = 0; j <= min(n, k); j++){
19            dp[n][j] = (dp[n][j] + dp[n - 2][j]) % mod;
20            dp[n][j] = (dp[n][j] + dp[n - 2][j - 1] * (n - j + 1)) % mod;
21        }
22    }
23    if(n == 1) return 1;
24    if(n == 2){
25        if(k == 0) return 1;
26        else if(k == 1 || k == 2) return 4;
27        else return 0;
28    }
29    for(int i = 0; i <= k; i++){
30        ans = (ans + dp[n][i] * dp[n - 1][k - i]) % mod;
31    }
32    return ans;
33 }
34 signed main(){
35     int n, k;
36     cin >> n >> k;
37     cout << DP(n, k) << "\n";
38     return 0;
39 }
```

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

Flow


```
1 #include <bits/stdc++.h>
2 #define pb push_back
3 using namespace std;
4 struct edge{
5     int u, v, f;
6 };
7 const int inf = 1000696969;
8 int cnt = 0;
9 array<int, 104> lvl, P;
10 array<edge, 10004> E;
11 array<array<char, 54>, 54> ans;
12 array<vector<int>, 104> G;
13 void add(int u, int v, int f){
14     E[cnt] = {u, v, f};
15     G[u].pb(cnt++);
16     E[cnt] = {v, u, 0};
17     G[v].pb(cnt++);
18 }
19 bool BFS(int s, int t){
20     int u;
21     queue<int> Q;
22     Q.push(s);
23     lvl[s] = 1;
24     while(!Q.empty()){
25         u = Q.front();
26         Q.pop();
27         for(int i : G[u]){
28             auto [o, v, f] = E[i];
29             if(lvl[v] || !f) continue;
30             lvl[v] = lvl[u] + 1;
31             Q.push(v);
32         }
33     }
34     return lvl[t];
35 }
36 int DFS(int u, int t, int f){
37     if(u == t || !f) return f;
38     int tmp, wut = 0;
39     for(int &i = P[u]; i < G[u].size(); i++){
40         auto &[eu, ev, ef] = E[G[u][i]];
41         auto &[bu, bv, bf] = E[G[u][i] ^ 1];
42         if(lvl[ev] - lvl[u] != 1) continue;
43         tmp = DFS(ev, t, min(f, ef));
44         ef -= tmp, bf += tmp;
45         f -= tmp, wut += tmp;
46     }
47     return wut;
48 }
49 int flow(int s, int t){
50     int wut = 0, tmp;
51     while(1){
52         for(int &l : lvl) l = 0;
53         if(!BFS(s, t)) break;
```

```

54     while(1){
55         for(int &p : P) p = 0;
56         if(tmp = DFS(s, t, inf)) wut += tmp;
57         else break;
58     }
59 }
60 return wut;
61 }
62 signed main(){
63     int n, w, in = 0, out = 0;
64     cin >> n;
65     for(int i = 1; i <= n; i++){
66         cin >> w;
67         in += w;
68         add(0, i, w);
69     }
70     for(int i = 1; i <= n; i++){
71         cin >> w;
72         out += w;
73         add(50 + i, 101, w);
74     }
75     for(int i = 1; i <= n; i++){
76         for(int j = 51; j <= 50 + n; j++){
77             add(i, j, 1);
78         }
79     }
80     if(flow(0, 101) != in || in != out){
81         cout << "-1\n";
82         return 0;
83     }
84     for(int u = 1; u <= n; u++){
85         for(int i : G[u]){
86             auto [o, v, f] = E[i];
87             if(v < 50) continue;
88             if(f) ans[u][v - 50] = '.';
89             else ans[u][v - 50] = 'X';
90         }
91     }
92     for(int i = 1; i <= n; i++){
93         for(int j = 1; j <= n; j++){
94             cout << ans[i][j];
95         }
96         cout << "\n";
97     }
98     return 0;
99 }
```

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

Min Cost Max Flow


```
1 #include <bits/stdc++.h>
2 #define pb push_back
3 using namespace std;
4 struct edge{
5     int u, v, f, c;
6 };
7 const int inf = 1000696969;
8 int cnt = 0;
9 array<bool, 104> in;
10 array<int, 104> dis, B;
11 array<edge, 10004> E;
12 array<array<char, 104>, 104> ans;
13 array<vector<int>, 104> G;
14 void add(int u, int v, int f, int c){
15     E[cnt] = {u, v, f, c};
16     G[u].pb(cnt++);
17     E[cnt] = {v, u, 0, -c};
18     G[v].pb(cnt++);
19 }
20 int SPFA(int s, int t){
21     int u;
22     queue<int> Q;
23     Q.push(s);
24     dis[s] = 0;
25     while(!Q.empty()){
26         u = Q.front();
27         Q.pop();
28         in[u] = 0;
29         for(int i : G[u]){
30             auto &[o, v, f, c] = E[i];
31             if(!f || dis[u] + c >= dis[v]) continue;
32             dis[v] = dis[u] + c;
33             B[v] = i;
34             if(in[v]) continue;
35             in[v] = 1;
36             Q.push(v);
37         }
38     }
39     return dis[t] == inf? 0 : 1;
40 }
41 void back(int u){
42     if(!u) return;
43     auto &[eu, ev, ef, ec] = E[B[u]];
44     auto &[bu, bv, bf, bc] = E[B[u] ^ 1];
45     ef--, bf++;
46     back(eu);
47 }
48 int flow(int s, int t){
49     int tmp, wut = 0;
50     while(1){
51         for(int &d : dis) d = inf;
52         for(int &b : B) b = 0;
53         for(bool &i : in) i = 0;
```

```
54         if(tmp == SPFA(s, t)) wut += tmp;
55     else break;
56     back(t);
57 }
58 return wut;
59 }
60 signed main(){
61     int n, w, out = 0, sum = 0;
62     cin >> n;
63     for(int i = 1; i <= n; i++){
64         cin >> w;
65         add(0, i, w, 0);
66     }
67     for(int i = 51; i <= 50 + n; i++){
68         cin >> w;
69         out += w;
70         add(i, 101, w, 0);
71     }
72     for(int i = 1; i <= n; i++){
73         for(int j = 51; j <= 50 + n; j++){
74             cin >> w;
75             add(i, j, 1, -w);
76         }
77     }
78     if(flow(0, 101) != out){
79         cout << "-1\n";
80         return 0;
81     }
82     for(int u = 1; u <= n; u++){
83         for(int i : G[u]){
84             auto [o, v, f, c] = E[i];
85             if(v < 50) continue;
86             if(f) ans[u][v - 50] = '.';
87             else{
88                 ans[u][v - 50] = 'X';
89                 sum -= c;
90             }
91         }
92     }
93     cout << sum << "\n";
94     for(int i = 1; i <= n; i++){
95         for(int j = 1; j <= n; j++){
96             cout << ans[i][j];
97         }
98         cout << "\n";
99     }
100    return 0;
101 }
```

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

DP

```
1 #include <bits/stdc++.h>
2 #define int long long
3 using namespace std;
4 const int mod = 1e9 + 7;
5 string S;
6 array<array<int, 504>, 504> vis;
7 array<array<int, 504>, 504> c, dp;
8 int C(int i, int j){
9     if(!j || i == j) return 1;
10    if(c[i][j]) return c[i][j];
11    return c[i][j] = (C(i - 1, j) + C(i - 1, j - 1)) % mod;
12 }
13 int DP(int i, int j){
14     if(i > j) return 1;
15     if(j - i == 1) return S[i] == S[j];
16     if((j - i) % 2 == 0) return 0;
17     if(vis[i][j]) return dp[i][j];
18     vis[i][j] = 1;
19     for(int k = i; k <= j; k++){
20         if(S[i] != S[k]) continue;
21         dp[i][j] += (DP(i + 1, k - 1) * DP(k + 1, j) % mod) * C((j - k) / 2, mod);
22         dp[i][j] %= mod;
23     }
24     return dp[i][j];
25 }
26 signed main(){
27     cin >> S;
28     cout << DP(0, S.size() - 1) << "\n";
29 }
30 }
```

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

DP


```
1 #include <bits/stdc++.h>
2 #define pb push_back
3 #define int long long
4 using namespace std;
5 struct trap{
6     int x, y;
7 };
8 const int mod = 1e9 + 7;
9 array<int, 1004> dp;
10 array<int, 2000004> fac;
11 vector<trap> T;
12 bool cmp(trap a, trap b){
13     if(a.x == b.x) return a.y < b.y;
14     return a.x < b.x;
15 }
16 void build(int n){
17     fac[0] = 1;
18     for(int i = 1; i <= n; i++){
19         fac[i] = i * fac[i - 1] % mod;
20     }
21 }
22 int exp(int x, int k){
23     int pro = 1;
24     for(int i = 1; i <= k; i <= 1){
25         if(i & k) pro = pro * x % mod;
26         x = x * x % mod;
27     }
28     return pro;
29 }
30 int C(int n, int k){
31     return (fac[n] * exp(fac[n - k], mod - 2) % mod) * exp(fac[k],
32 }
33 int DP(int n){
34     for(int i = 0; i <= n; i++){
35         auto [x, y] = T[i];
36         dp[i] = C(x + y - 2, x - 1);
37         for(int j = 0; j < i; j++){
38             auto [tx, ty] = T[j];
39             if(y < ty) continue;
40             dp[i] = (dp[i] - dp[j] * C(x - tx + y - ty, x - tx) % m
41         }
42     }
43     return dp[n];
44 }
45 signed main(){
46     int n, m, x, y;
47     cin >> n >> m;
48     build(2 * n);
49     for(int i = 1; i <= m; i++){
50         cin >> x >> y;
51         T.pb({x, y});
52     }
53     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


```
1 #include <bits/stdc++.h>
2 #define int long long
3 #define cpx complex<double>
4 #define i imag
5 #define r real
6 using namespace std;
7 const int N = 1 << 19;
8 const double pi = acos(-1);
9 array<cpx, 1 << 19> A, B, C, X;
10 cpx ei(double x){
11     return {cos(x), sin(x)};
12 }
13 void FFT(array<cpx, 1 << 19> &F){
14     int n;
15     cpx x;
16     for(int i = 0, j = 0; i < N; i++){
17         if(i > j) swap(F[i], F[j]);
18         for(int k = N >> 1; (j ^= k) < k; k >>= 1);
19     }
20     for(int k = 2; k <= N; k <= 1){
21         n = k >> 1;
22         for(int j = 0; j < N; j += k){
23             for(int i = j; i < j + n; i++){
24                 x = X[(i - j) * N / k] * F[i + n];
25                 F[i + n] = F[i] - x;
26                 F[i] += x;
27             }
28         }
29     }
30 }
31 int rnd(double x){
32     double z = (int)x;
33     if(x - z >= 0.5) return (int)z + 1;
34     else return (int)z;
35 }
36 signed main(){
37     int n, pre = 0, zero = 0, cnt = 0;
38     string S;
39     cin >> S;
40     n = S.size();
41     B[n] = {1, 0};
42     for(char s : S){
43         if(s == '1'){
44             pre++;
45             zero += cnt * (cnt + 1) / 2;
46             cnt = 0;
47         }else cnt++;
48         A[pre] += 1;
49         B[n - pre] += 1;
50     }
51     zero += cnt * (cnt + 1) / 2;
52     for(int i = 0; i < N; i++){
53         X[i] = ei(2.0 * pi * i / N);
```

```
54     }
55     FFT(A), FFT(B);
56     for(int i = 0; i < N; i++){
57         C[i] = A[i] * B[i];
58         X[i] = conj(X[i]);
59     }
60     FFT(C);
61     cout << zero << " ";
62     for(int i = n + 1; i <= 2 * n; i++){
63         cout << rnd(C[i].r()) / (double)N) << " ";
64     }
65     cout << "\n";
66     return 0;
67 }
```

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

Treap


```
1 #include <bits/stdc++.h>
2 #define pb push_back
3 using namespace std;
4 const int inf = 1e9;
5 struct treap{
6     int val, man, siz, pri;
7     bool rev;
8     treap *lc, *rc;
9     treap(int v){
10         val = man = v;
11         siz = 1;
12         pri = rand();
13         rev = 0;
14         lc = rc = nullptr;
15     }
16     void pull(){
17         man = min(min(lc? lc->man : inf, rc? rc->man : inf), val);
18         siz = (lc? lc->siz : 0) + (rc? rc->siz : 0) + 1;
19     }
20     void push(){
21         if(!rev) return;
22         swap(lc, rc);
23         if(lc) lc->rev ^= 1;
24         if(rc) rc->rev ^= 1;
25         rev = 0;
26     }
27     int find(int k){
28         push();
29         int ls = (lc? lc->siz : 0) + 1;
30         if(val == k) return ls;
31         if(lc && lc->man == k) return lc->find(k);
32         else return rc->find(k) + ls;
33     }
34 };
35 vector<pair<int, int>> ans;
36 int size(treap *t){
37     return t? t->siz : 0;
38 }
39 treap* merge(treap *a, treap *b){
40     if(!a || !b) return a? a : b;
41     if(a->pri > b->pri){
42         a->push();
43         a->rc = merge(a->rc, b);
44         a->pull();
45         return a;
46     }else{
47         b->push();
48         b->lc = merge(a, b->lc);
49         b->pull();
50         return b;
51     }
52 }
53 void split(treap *t, treap *&a, treap *&b, int k){
```

```
54     if(!t){
55         a = b = nullptr;
56         return;
57     }
58     t->push();
59     if(k <= size(t->lc)){
60         b = t;
61         split(t->lc, a, b->lc, k);
62         b->pull();
63     }else{
64         a = t;
65         split(t->rc, a->rc, b, k - size(t->lc) - 1);
66         a->pull();
67     }
68 }
69 signed main(){
70     srand(time(NULL));
71     int n, x, p;
72     treap *t = nullptr, *a = nullptr, *b = nullptr, *c = nullptr;
73     cin >> n;
74     for(int i = 1; i <= n; i++){
75         cin >> x;
76         t = merge(t, new treap(x));
77     }
78     for(int i = 1; i <= n; i++){
79         split(t, a, b, i - 1);
80         p = b->find(i);
81         if(p == 1){
82             t = merge(a, b);
83             continue;
84         }
85         ans.pb({i, i + p - 1});
86         split(b, b, c, p);
87         b->rev ^= 1;
88         t = merge(a, merge(b, c));
89     }
90     cout << ans.size() << "\n";
91     for(auto [l, r] : ans) cout << l << " " << r << "\n";
92     return 0;
93 }
```

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

1 啾啾待哺

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

DP

```
1 #include <bits/stdc++.h>
2 #define int long long
3 using namespace std;
4 int l, r;
5 array<int, 104> H, S, K;
6 array<int, 100004> Q;
7 array<array<int, 100004>, 2> dp;
8 int DP(int n, int x){
9     int p = 0, ans = 0;
10    for(int i = 1; i <= n; i++){
11        p ^= 1;
12        for(int j = 0; j < H[i]; j++){
13            l = 0, r = -1;
14            for(int k = j; k <= x; k += H[i]){
15                if(l <= r && Q[l] + K[i] * H[i] < k) l++;
16                while(l <= r && dp[p ^ 1][k] >= dp[p ^ 1][Q[r]] + (Q[++r] = k);
17                dp[p][k] = dp[p ^ 1][Q[l]] + (k - Q[l]) / H[i] * S[i];
18                ans = max(ans, dp[p][k]);
19            }
20        }
21    }
22    return ans;
23 }
24 }
25 signed main(){
26     int n, x;
27     cin >> n >> x;
28     for(int i = 1; i <= n; i++) cin >> H[i];
29     for(int i = 1; i <= n; i++) cin >> S[i];
30     for(int i = 1; i <= n; i++) cin >> K[i];
31     cout << DP(n, x) << "\n";
32     return 0;
33 }
```

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

DSU

```
1 #include <bits/stdc++.h>
2 #define pb push_back
3 using namespace std;
4 int com;
5 array<int, 100004> DSU;
6 vector<int> ans;
7 vector<pair<int, int>> B;
8 set<pair<int, int>> E;
9 int find(int u){
10     if(DSU[u] == u) return u;
11     return DSU[u] = find(DSU[u]);
12 }
13 void onion(int u, int v){
14     int U = find(u), V = find(v);
15     if(U == V) return;
16     DSU[V] = U;
17     com--;
18 }
19 void run(int k){
20     for(auto [a, b] : E){
21         onion(a, b);
22     }
23     ans.pb(com);
24     for(auto [a, b] : B){
25         onion(a, b);
26         ans.pb(com);
27     }
28     ans.pop_back();
29 }
30 signed main(){
31     int n, m, k, a, b;
32     cin >> n >> m >> k;
33     com = n;
34     for(int i = 1; i <= n; i++) DSU[i] = i;
35     while(m--){
36         cin >> a >> b;
37         if(a > b) swap(a, b);
38         E.insert({a, b});
39     }
40     for(int i = 0; i < k; i++){
41         cin >> a >> b;
42         if(a > b) swap(a, b);
43         E.erase({a, b});
44         B.pb({a, b});
45     }
46     reverse(B.begin(), B.end());
47     run(k);
48     reverse(ans.begin(), ans.end());
49     for(int c : ans) cout << c << " ";
50     cout << "\n";
51     return 0;
52 }
```

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

Dijkstra Dominator Tree


```
1 #include <bits/stdc++.h>
2 #define pb push_back
3 #define int long long
4 using namespace std;
5 struct edge{
6     int v, w;
7 };
8 struct cmp{
9     bool operator()(edge a, edge b){
10         return a.w > b.w;
11     }
12 };
13 int k = 0;
14 array<int, 100004> dis, P, rev, L, sdom, idom, pre, DSU;
15 array<vector<int>, 100004> D, R, buk;
16 array<vector<edge>, 100004> G;
17 vector<int> ans;
18 int find(int u, int x = 0){
19     if(u == DSU[u]) return x? -1 : u;
20     int v = find(DSU[u], x + 1);
21     if(v < 0) return u;
22     if(sdom[L[DSU[u]]] < sdom[L[u]]) L[u] = L[DSU[u]];
23     DSU[u] = v;
24     return x? v : L[u];
25 }
26 void onion(int u, int v){
27     DSU[v] = u;
28 }
29 void run(int s, int n){
30     priority_queue<edge, vector<edge>, cmp> Q;
31     Q.push({s, 1});
32     while(!Q.empty()){
33         auto [u, d] = Q.top();
34         Q.pop();
35         if(dis[u]) continue;
36         dis[u] = d;
37         for(auto [v, w] : G[u]){
38             Q.push({v, d + w});
39         }
40     }
41     for(int u = 1; u <= n; u++){
42         for(auto [v, w] : G[u]){
43             if(dis[u] + w == dis[v]) D[u].pb(v);
44         }
45     }
46 }
47 void DFS(int u){
48     P[u] = ++k, rev[k] = u, L[k] = k, sdom[k] = k, DSU[k] = k;
49     for(int v : D[u]){
50         if(!P[v]){
51             DFS(v);
52             pre[P[v]] = P[u];
53         }
54     }
55 }
```

```

54         R[P[v]].pb(P[u]);
55     }
56 }
57 void DOM(int n){
58     int w;
59     for(int u = n; u; u--){
60         for(int v : R[u]){
61             sdom[u] = min(sdom[u], sdom[find(v)]);
62         }
63         if(u > 1) buk[sdom[u]].pb(u);
64         for(int v : buk[u]){
65             w = find(v);
66             if(sdom[v] == sdom[w]) idom[v] = sdom[v];
67             else idom[v] = w;
68         }
69         onion(pre[u], u);
70     }
71     for(int u = 2; u <= n; u++){
72         if(sdom[u] != idom[u]) idom[u] = idom[idom[u]];
73     }
74 }
75 void back(int u){
76     ans.pb(rev[u]);
77     if(u == 1) return;
78     back(idom[u]);
79 }
80 signed main(){
81     int n, m, a, b, c;
82     cin >> n >> m;
83     while(m--){
84         cin >> a >> b >> c;
85         G[a].pb({b, c});
86     }
87     run(1, n);
88     DFS(1);
89     DOM(n);
90     back(P[n]);
91     sort(ans.begin(), ans.end());
92     cout << ans.size() << "\n";
93     for(int u : ans) cout << u << " ";
94     cout << "\n";
95     return 0;
96 }

```

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

Doubling RMQ


```
1 #include <bits/stdc++.h>
2 #define pb push_back
3 #define int long long
4 using namespace std;
5 struct ooo{
6     int l, r, t;
7 };
8 array<int, 200004> X, ans, P;
9 array<int, 400004> S;
10 array<array<int, 200004>, 20> T;
11 vector<ooo> Q;
12 int highbit(int x){
13     int p = 0;
14     for(int i = 4; i >= 0; i--){
15         if(1 << (p + (1 << i)) <= x) p += (1 << i);
16     }
17     return p;
18 }
19 void build(int n){
20     for(int i = 1; 1 << i <= n; i++){
21         for(int j = 1; j <= n; j++){
22             T[i][j] = min(T[i - 1][j], T[i - 1][j + (1 << (i - 1))]);
23         }
24     }
25 }
26 void find(int n){
27     int h;
28     for(int k = 1, p = 1; k < 1 << 30; k <= 1, p = 1){
29         S[1] = 0, T[0][1] = 1ll << 60;
30         for(int i = 1; i <= n; i++){
31             if(X[i] >= k && X[i] < k << 1){
32                 P[i] = ++p;
33                 S[p] = X[i] + S[p - 1];
34                 T[0][p] = X[i];
35                 p++;
36             } else P[i] = p;
37             S[p] = S[p - 1], T[0][p] = 1ll << 60;
38         }
39         build(p);
40         for(auto [l, r, t] : Q){
41             l = P[l], r = P[r], h = highbit(r - l + 1);
42             if(ans[t] >= min(T[h][l], T[h][r + 1 - (1 << h)])){
43                 ans[t] += S[r] - S[l - 1];
44             }
45         }
46     }
47 }
48 signed main(){
49     int n, q, l, r;
50     cin >> n >> q;
51     for(int i = 1; i <= n; i++) cin >> X[i];
52     for(int i = 1; i <= q; i++){
53         cin >> l >> r;
```

```
54     Q.pb({l, r, i});
55     ans[i] = 1;
56 }
57 find(n);
58 for(int i = 1; i <= q; i++) cout << ans[i] << "\n";
59 return 0;
60 }
```

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

通靈

```
1 #include <bits/stdc++.h>
2 using namespace std;
3 signed main(){
4     int x, y;
5     cin >> x >> y;
6     cout << ((x - 1) ^ (y - 1)) << "\n";
7     return 0;
8 }
```

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

差分


```
1 #include <bits/stdc++.h>
2 #define int long long
3 using namespace std;
4 array<int, 1004> H;
5 array<array<char, 1004>, 1004> G;
6 array<array<int, 1004>, 1004> cnt;
7 void run(int n, int m){
8     int now;
9     for(int i = 1; i <= n; i++){
10         stack<pair<int, int>> S;
11         S.push({0, 0});
12         for(int j = 1; j <= m; j++){
13             now = j;
14             if(G[i][j] == '*') H[j] = 0;
15             else H[j]++;
16             while(!S.empty()){
17                 auto [h, p] = S.top();
18                 if(H[j] < h){
19                     cnt[h][j - p]++;
20                     S.pop();
21                     now = p;
22                     auto [h2, p2] = S.top();
23                     cnt[max(h2, H[j])][j - p]--;
24                 }else{
25                     break;
26                 }
27             }
28             S.push({H[j], now});
29         }
30     }
31     for(int i = n; i; i--){
32         for(int j = m; j; j--){
33             cnt[i][j] += cnt[i][j + 1];
34         }
35     }
36     for(int i = n; i; i--){
37         for(int j = m; j; j--){
38             cnt[i][j] += cnt[i + 1][j] + cnt[i][j + 1] - cnt[i + 1]
39         }
40     }
41 }
42 signed main(){
43     int n, m;
44     cin >> n >> m;
45     for(int i = 1; i <= n; i++){
46         for(int j = 1; j <= m; j++){
47             cin >> G[i][j];
48         }
49         G[i][m + 1] = '*';
50     }
51     run(n, m + 1);
52     for(int i = 1; i <= n; i++){
53         for(int j = 1; j <= m; j++)
```

```
54         cout << cnt[i][j] << " ";
55     }
56     cout << "\n";
57 }
58 return 0;
59 }
```

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

1 啾啾待哺

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

Greedy

```
1 #include <bits/stdc++.h>
2 #define int long long
3 using namespace std;
4 array<int, 200004> L;
5 int div(int n){
6     int a, b, sum = 0;
7     priority_queue<int, vector<int>, greater<int>> Q;
8     for(int i = 0; i < n; i++) Q.push(L[i]);
9     while(Q.size() > 1){
10         a = Q.top(), Q.pop();
11         b = Q.top(), Q.pop();
12         sum += a + b;
13         Q.push(a + b);
14     }
15     return sum;
16 }
17 signed main(){
18     int x, n;
19     cin >> x >> n;
20     for(int i = 0; i < n; i++){
21         cin >> L[i];
22     }
23     cout << div(n) << "\n";
24     return 0;
25 }
```

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

DP

```
1 #include <bits/stdc++.h>
2 #define int long long
3 using namespace std;
4 const int mod = 1e9 + 7;
5 array<int, 104> T;
6 array<array<array<int, 20004>, 104>, 2> dp;
7 int DP(int n, int x, int sum){
8     int p = 0, ans = 0;
9     dp[0][0][sum] = 1;
10    for(int i = 1; i <= n; i++){
11        p ^= 1;
12        for(int j = 0; j <= i; j++){
13            for(int k = -sum; k <= sum; k++){
14                dp[p][j][k + sum] = 0;
15                if(j && k + T[i] <= sum) dp[p][j][k + sum] += dp[p]
16                dp[p][j][k + sum] += (j + 1) * dp[p ^ 1][j][k + sum];
17                if(k - T[i] >= -sum) dp[p][j][k + sum] += (j + 1) *
18                dp[p][j][k + sum] %= mod;
19            }
20        }
21    }
22    for(int i = 0; i <= x; i++){
23        ans = (ans + dp[n & 1][0][i + sum]) % mod;
24    }
25    return ans;
26 }
27 signed main(){
28     int n, x, sum = 0;
29     cin >> n >> x;
30     for(int i = 1; i <= n; i++){
31         cin >> T[i];
32         sum += T[i];
33     }
34     sort(T.begin() + 1, T.begin() + n + 1);
35     cout << DP(n, x, sum) << "\n";
36     return 0;
37 }
```

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

Topological Sort

```
1 #include <bits/stdc++.h>
2 #define pb push_back
3 using namespace std;
4 bitset<100004> vis;
5 array<int, 100004> in;
6 array<vector<pair<int, int>>, 100004> G;
7 vector<int> V;
8 void DFS(int u){
9     if(vis[u]) return;
10    vis[u] = 1;
11    V.pb(u);
12    for(auto [v, d] : G[u]){
13        if(d) in[v]++;
14        DFS(v);
15    }
16 }
17 int topu(int u){
18     if(vis[u]) return 0;
19     int nc = 0;
20     V.clear();
21     DFS(u);
22     queue<int> Q;
23     for(int v : V){
24         if(!in[v]) Q.push(v);
25     }
26     while(!Q.empty()){
27         u = Q.front();
28         Q.pop();
29         nc++;
30         for(auto [v, d] : G[u]){
31             if(!d) continue;
32             in[v]--;
33             if(!in[v]) Q.push(v);
34         }
35     }
36     return nc == V.size()? V.size() - 1 : V.size();
37 }
38 signed main(){
39     int n, m, a, b, edge = 0;
40     cin >> n >> m;
41     while(m--){
42         cin >> a >> b;
43         G[a].pb({b, 1});
44         G[b].pb({a, 0});
45     }
46     for(int i = 1; i <= n; i++){
47         edge += topu(i);
48     }
49     cout << edge << "\n";
50     return 0;
51 }
```

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

1 | 噗嗷待哺

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

Tree Centriod Hash


```
1 #include <bits/stdc++.h>
2 #define int long long
3 #define pb push_back
4 using namespace std;
5 const int k = 41, mod = 1000696969;
6 int n;
7 array<int, 100004> H, S1, S2;
8 array<vector<int>, 100004> T1, T2;
9 vector<int> C1, C2;
10 void build(){
11     H[0] = 1;
12     for(int i = 1; i < 100004; i++){
13         H[i] = k * H[i - 1] % mod;
14     }
15 }
16 int DFS(array<vector<int>, 100004> &T, vector<int> &C, int u, int p,
17         int tmp, s = 1;
18         bool cen = 1;
19         for(int v : T[u]){
20             if(v == pre) continue;
21             tmp = DFS(T, C, v, u);
22             if(2 * tmp > n) cen = 0;
23             s += tmp;
24         }
25         if(2 * s < n) cen = 0;
26         if(cen) C.pb(u);
27         return s;
28     }
29     int HAS(array<vector<int>, 100004> &T, array<int, 100004> &S, int u,
30         int sum = 0;
31         S[u] = 0;
32         vector<pair<int, int>> has;
33         for(int v : T[u]){
34             if(v == pre) continue;
35             has.pb({HAS(T, S, v, u), S[v]});
36         }
37         sort(has.begin(), has.end());
38         for(auto [h, s] : has){
39             sum = (sum + H[S[u]] * h) % mod;
40             S[u] += s;
41         }
42         sum = (sum + H[S[u]] * (S[u] + 1)) % mod;
43         S[u]++;
44         return sum;
45     }
46     signed main(){
47         int t, a, b;
48         bool ok;
49         cin >> t;
50         build();
51         while(t--){
52             cin >> n;
53             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


```
1 #include <bits/stdc++.h>
2 #define int long long
3 #define pb push_back
4 using namespace std;
5 struct edge{
6     int v, t;
7 };
8 int k = 0, r = 0, o = 0;
9 bitset<200004> nee, vis;
10 array<int, 100004> P, low, bcc;
11 array<int, 200004> in, out, ecc;
12 array<int, 400004> BIT;
13 array<array<int, 20>, 200004> A;
14 array<vector<int>, 200004> T;
15 array<vector<edge>, 100004> G;
16 stack<int> S;
17 set<int> E;
18 void update(int p, int x){
19     for(; p < 400004; p += p & -p) BIT[p] += x;
20 }
21 int query(int p){
22     int sum = 0;
23     for(; p; p -= p & -p) sum += BIT[p];
24     return sum;
25 }
26 int hash(int u, int v){
27     if(u > v) swap(u, v);
28     return u * 100001 + v;
29 }
30 void DFST(int u, int pre){
31     int cnt = 0, s;
32     P[u] = low[u] = ++r;
33     for(auto [v, t] : G[u]){
34         if(v == pre) continue;
35         if(!vis[t]){
36             S.push(t);
37             vis[t] = 1;
38         }
39         if(P[v]) low[u] = min(low[u], P[v]);
40         else{
41             cnt++;
42             DFST(v, u);
43             low[u] = min(low[u], low[v]);
44             if(low[v] >= P[u]){
45                 k++, nee[u] = 1;
46                 while(!S.empty()){
47                     s = S.top(), S.pop();
48                     ecc[s] = k;
49                     if(s == t) break;
50                 }
51             }
52         }
53     }
}
```

```
54         if(u == 1 && cnt < 2) nee[1] = 0;
55     }
56 void DFS(int u, int pre){
57     in[u] = ++o;
58     for(int v : T[u]){
59         if(v == pre) continue;
60         DFS(v, u);
61         A[v][0] = u;
62     }
63     out[u] = ++o;
64 }
65 void DAB0(int n){
66     in[0] = 0, out[0] = 1 << 20;
67     for(int j = 1; j < 20; j++){
68         for(int i = 1; i <= n; i++){
69             A[i][j] = A[A[i][j - 1]][j - 1];
70         }
71     }
72 }
73 int LCA(int a, int b){
74     for(int i = 19; i >= 0; i--){
75         if(in[A[a][i]] > in[b] || out[A[a][i]] < out[b]) a = A[a][i];
76     }
77     if(in[a] > in[b] || out[a] < out[b]) a = A[a][0];
78     return a;
79 }
80 signed main(){
81     int n, m, q, a, b, c, lca, ans;
82     cin >> n >> m >> q;
83     for(int i = 1; i <= m; i++){
84         cin >> a >> b;
85         if(E.find(hesh(a, b)) != E.end() || a == b) continue;
86         E.insert(hesh(a, b));
87         G[a].pb({b, i});
88         G[b].pb({a, i});
89     }
90     DFST(1, 0);
91     for(int i = 1; i <= n; i++){
92         if(nee[i]) bcc[i] = ++k;
93         else bcc[i] = ecc[G[i][0].t];
94     }
95     for(int u = 1; u <= n; u++){
96         if(!nee[u]) continue;
97         for(auto [v, t] : G[u]){
98             T[bcc[u]].pb(ecc[t]);
99             T[ecc[t]].pb(bcc[u]);
100        }
101    }
102    for(int i = 1; i <= k; i++){
103        sort(T[i].begin(), T[i].end());
104        T[i].erase(unique(T[i].begin(), T[i].end()), T[i].end());
105    }
106    DFS(1, 0);
107 }
```

```
107 // DABU(K);
108 while(q--){
109     cin >> a >> b >> c;
110     if(a == c || b == c){
111         cout << "NO\n";
112         continue;
113     }else if(!nee[c]){
114         cout << "YES\n";
115         continue;
116     }
117     a = bcc[a], b = bcc[b], c = bcc[c];
118     update(in[c], 1), update(out[c], -1);
119     lca = LCA(a, b);
120     if(in[a] > in[b]) swap(a, b);
121     if(c == lca) ans = 1;
122     else if(a == lca) ans = query(in[b]) - query(in[a] - 1);
123     else ans = query(in[b]) - query(out[a] - 1);
124     update(in[c], -1), update(out[c], 1);
125     cout << (ans? "NO\n" : "YES\n");
126 }
127 return 0;
128 }
```

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

Segment Tree


```
1 #include <bits/stdc++.h>
2 #define pb push_back
3 #define int long long
4 #define mid ((l + r) >> 1)
5 #define lc (p << 1)
6 #define rc ((p << 1) | 1)
7 using namespace std;
8 struct ooo{
9     int x, l, r, v;
10 };
11 const int inf = 1e6;
12 array<int, 8000004> man, tag, cnt;
13 vector<ooo> Q;
14 bool cmp(ooo a, ooo b){
15     return a.x < b.x;
16 }
17 void pull(int p){
18     man[p] = min(man[lc], man[rc]);
19     if(man[lc] < man[rc]) cnt[p] = cnt[lc];
20     else if(man[rc] < man[lc]) cnt[p] = cnt[rc];
21     else cnt[p] = cnt[lc] + cnt[rc];
22 }
23 void push(int p){
24     man[lc] += tag[p];
25     man[rc] += tag[p];
26     tag[lc] += tag[p];
27     tag[rc] += tag[p];
28     tag[p] = 0;
29 }
30 void build(int p, int l, int r){
31     if(l == r){
32         cnt[p] = 1;
33         return;
34     }
35     build(lc, l, mid);
36     build(rc, mid + 1, r);
37     pull(p);
38 }
39 void update(int p, int l, int r, int ql, int qr, int x){
40     if(ql > r || qr < l) return;
41     if(ql <= l && qr >= r){
42         man[p] += x;
43         tag[p] += x;
44         return;
45     }
46     push(p);
47     update(lc, l, mid, ql, qr, x);
48     update(rc, mid + 1, r, ql, qr, x);
49     pull(p);
50 }
51 signed main(){
52     int n, x1, y1, x2, y2, p = 0, sum = 0;
53     cin >> n;
```

```
54     for(int i = 1; i <= n; i++){
55         cin >> x1 >> y1 >> x2 >> y2;
56         Q.pb({x1, y1, y2 - 1, 1});
57         Q.pb({x2, y1, y2 - 1, -1});
58     }
59     sort(Q.begin(), Q.end(), cmp);
60     build(1, -inf, inf);
61     for(int i = -inf; i < inf; i++){
62         while(p < Q.size() && Q[p].x == i){
63             auto [x, l, r, v] = Q[p++];
64             update(1, -inf, inf, l, r, v);
65         }
66         sum += 2 * inf + 1 - cnt[1];
67     }
68     cout << sum << "\n";
69     return 0;
70 }
```

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

1 噰嗷待哺

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

重心剖分


```
1 #include <bits/stdc++.h>
2 #define pb push_back
3 using namespace std;
4 array<int, 200004> S, M, D, F, N;
5 array<array<int, 20>, 200004> dis;
6 array<vector<int>, 200004> T;
7 vector<int> V, ans;
8 vector<pair<int, int>> Q;
9 int DFS(int u){
10     if(S[u]) return 0;
11     int s;
12     S[u] = 1, V.pb(u);
13     for(int v : T[u]){
14         s = DFS(v);
15         M[u] = max(M[u], s), S[u] += s;
16     }
17     return S[u];
18 }
19 void walk(int u, int dep, int s){
20     if(S[u]) return;
21     S[u] = 1, dis[u][dep] = s;
22     for(int v : T[u]){
23         walk(v, dep, s + 1);
24     }
25     S[u] = 0;
26 }
27 int CUT(int u, int dep){
28     V.clear();
29     DFS(u);
30     int cen, n = V.size();
31     for(int v : V){
32         if(2 * M[v] <= n && 2 * S[v] >= n) cen = v;
33         S[v] = M[v] = 0;
34     }
35     walk(cen, dep, 0);
36     D[cen] = dep, S[cen] = 1;
37     for(int v : T[cen]){
38         if(!S[v]) F[CUT(v, dep + 1)] = cen;
39     }
40     return cen;
41 }
42 void RUN(int u, int pre, int dep){
43     Q.pb({dep, u});
44     for(int v : T[u]){
45         if(v == pre) continue;
46         RUN(v, u, dep + 1);
47     }
48 }
49 void update(int u, int v){
50     if(!u) return;
51     N[u] = min(N[u], dis[v][D[u]]);
52     update(F[u], v);
53 }
```

```
54     int query(int u, int v){  
55         if(!u) return 1 << 20;  
56         return min(N[u] + dis[v][D[u]], query(F[u], v));  
57     }  
58     signed main(){  
59         cin.tie(0), cout.tie(0), ios::sync_with_stdio(0);  
60         int n, d, a, b;  
61         cin >> n >> d;  
62         for(int i = 1; i < n; i++){  
63             cin >> a >> b;  
64             T[a].pb(b);  
65             T[b].pb(a);  
66         }  
67         CUT(1, 0);  
68         for(int &s : S) s = 0;  
69         for(int &s : N) s = 1 << 20;  
70         RUN(1, 0, 0);  
71         sort(Q.begin(), Q.end());  
72         reverse(Q.begin(), Q.end());  
73         for(auto [s, u] : Q){  
74             if(query(u, u) < d) continue;  
75             ans.pb(u);  
76             update(u, u);  
77         }  
78         cout << ans.size() << "\n";  
79         for(int u : ans) cout << u << " ";  
80         cout << "\n";  
81         return 0;  
82     }
```

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

DP

```
1 #include <bits/stdc++.h>
2 #define int long long
3 using namespace std;
4 const int mod = 1e9 + 7;
5 array<array<array<int, 2>, 1004>, 1004> dp;
6 int DP(int n){
7     dp[1][0][0] = 1;
8     dp[2][1][1] = 2;
9     for(int i = 3; i <= n; i++){
10         for(int j = 0; j < i; j++){
11             dp[i][j][0] += (i - j - 2) * dp[i - 1][j][0];
12             dp[i][j][0] += (i - j - 1) * dp[i - 1][j][1];
13             dp[i][j][0] += (j + 1) * dp[i - 1][j + 1][0];
14             dp[i][j][0] += j * dp[i - 1][j + 1][1];
15             if(j) dp[i][j][1] += 2 * dp[i - 1][j - 1][0];
16             dp[i][j][1] += dp[i - 1][j][1];
17             if(j) dp[i][j][1] += dp[i - 1][j - 1][1];
18             dp[i][j][0] %= mod, dp[i][j][1] %= mod;
19         }
20     }
21     return dp[n][0][0];
22 }
23 signed main(){
24     int n;
25     cin >> n;
26     cout << DP(n) << "\n";
27     return 0;
28 }
```

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

Caley's Formula

```
1 #include <bits/stdc++.h>
2 #define int long long
3 using namespace std;
4 const int mod = 1e9 + 7;
5 array<int, 5004> N;
6 array<array<int, 5004>, 5004> C, S;
7 void build(int n){
8     N[0] = S[0][0] = C[0][0] = 1;
9     for(int i = 1; i <= n; i++){
10         N[i] = n * N[i - 1] % mod;
11         C[i][0] = 1;
12         for(int j = 1; j <= i; j++){
13             C[i][j] = (C[i - 1][j] + C[i - 1][j - 1]) % mod;
14             S[i][j] = ((i - 1) * S[i - 1][j] + S[i - 1][j - 1]) % m
15         }
16     }
17 }
18 signed main(){
19     int n, sum = 0;
20     cin >> n;
21     build(n);
22     for(int i = 1; i <= n; i++, sum = 0){
23         for(int j = 1; j < n; j++){
24             sum = (sum + ((C[n][j] * S[j][i]) % mod) * ((j * N[n - 1] + 1) * N[i - 1])) % mod;
25         }
26         sum = (sum + S[n][i]) % mod;
27         cout << sum << "\n";
28     }
29     return 0;
30 }
```

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

SCC Maximal Flow


```
1 #include <bits/stdc++.h>
2 #define pb push_back
3 using namespace std;
4 const int t = 100001;
5 int k = 0;
6 bitset<100004> vis, F;
7 array<int, 100004> in, out, scc, S;
8 array<vector<int>, 100004> G, R;
9 array<vector<pair<int, int>>, 100004> D;
10 vector<int> ord, I, O, A, B;
11 vector<pair<int, int>> ans;
12 void BFS(int u){
13     if(vis[u]) return;
14     vis[u] = 1;
15     for(int v : R[u]) BFS(v);
16     ord.pb(u);
17 }
18 void DFS(int u){
19     if(scc[u]) return;
20     scc[u] = k;
21     for(int v : G[u]) DFS(v);
22 }
23 bool flow(int u){
24     if(u == t) return 1;
25     for(auto &[v, f] : D[u]){
26         if(f) continue;
27         f = 1;
28         if(flow(v)){
29             if(!in[u]) I.pb(u), F[u] = 1;
30             if(!out[u]) O.pb(u), F[u] = 1;
31             return 1;
32         }
33     }
34     return 0;
35 }
36 signed main(){
37     int n, m, a, b, s;
38     cin >> n >> m;
39     while(m--){
40         cin >> a >> b;
41         G[a].pb(b);
42         R[b].pb(a);
43     }
44     for(int i = 1; i <= n; i++) BFS(i);
45     reverse(ord.begin(), ord.end());
46     for(int u : ord){
47         if(!scc[u]) k++;
48         DFS(u);
49     }
50     if(k == 1){
51         cout << "0\n";
52         return 0;
53     }
```

```
54     for(int u = 1; u <= n; u++){
55         S[scc[u]] = u;
56         for(int v : G[u]){
57             if(scc[v] == scc[u]) continue;
58             D[scc[u]].pb({scc[v], 0});
59         }
60     }
61     for(int i = 1; i <= k; i++){
62         D[i].erase(unique(D[i].begin(), D[i].end()), D[i].end());
63         for(auto [j, f] : D[i]){
64             in[j]++, out[i]++;
65         }
66     }
67     for(int i = 1; i <= k; i++){
68         if(!out[i]) D[i].pb({t, 0});
69     }
70     for(int i = 1; i <= k; i++){
71         if(!in[i]) flow(i), a = i;
72         if(!out[i]) b = i;
73     }
74     for(int i = 1; i <= k; i++){
75         if(F[i]) continue;
76         if(!in[i]) A.pb(i);
77         if(!out[i]) B.pb(i);
78     }
79     for(int i = 0; i < I.size() - 1; i++){
80         ans.pb({0[i], I[i + 1]});
81     }
82     if(I.size()) ans.pb({0.back(), I[0]});
83     for(int i = 0; i < min(A.size(), B.size()); i++){
84         ans.pb({B[i], A[i]});
85     }
86     if(A.size() > B.size()){
87         for(int i = B.size(); i < A.size(); i++){
88             ans.pb({b, A[i]});
89         }
90     }else{
91         for(int i = A.size(); i < B.size(); i++){
92             ans.pb({B[i], a});
93         }
94     }
95     cout << ans.size() << "\n";
96     for(auto [u, v] : ans) cout << S[u] << " " << S[v] << "\n";
97     return 0;
98 }
```

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

1 啾啾待哺

