

CSES PLAN III

 [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 III

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

目錄

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

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

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

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

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

- Advanced Techniques

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

- Additional Problems

Advanced Techniques

[Meet in the Middle](#) (<https://cses.fi/problemset/task/1628>).

折半列舉 二分搜

```
1  #include <iostream>
2  #include <array>
3  #include <unordered_map>
4  #include <algorithm>
5  #define int long long
6  using namespace std;
7  array<int, 40> T;
8  array<int, 1 << 20> F, S;
9  void ju(int s, int t, array<int, 1 << 20> &A){
10     for(int i = 0; i < 1 << (t - s); i++){
11         for(int j = 0; j < (t - s); j++){
12             if(i & (1 << j)){
13                 A[i] = T[s + j] + A[i ^ (1 << j)];
14                 break;
15             }
16         }
17     }
18 }
19 signed main(){
20     int n, x, s, cnt = 0;
21     cin >> n >> x;
22     for(int i = 0; i < n; i++){
23         cin >> T[i];
24     }
25     ju(0, n / 2, F);
26     ju(n / 2, n, S);
27     s = 1 << (n - n / 2);
28     sort(S.begin(), S.begin() + s);
29     for(int f : F){
30         if(f) cnt += upper_bound(S.begin(), S.begin() + s, x - f) -
31     }
32     cnt += upper_bound(S.begin(), S.begin() + s, x) - lower_bound(S
33     cout << cnt;
34     return 0;
35 }
```

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

壓常

```
1  #include <iostream>
2  #include <array>
3  #include <bitset>
4  using namespace std;
5  array<int, 20004> S;
6  signed main(){
7      cin.tie(0), cout.tie(0), ios::sync_with_stdio(0);
8      int n, k, ans = 30;
9      char b;
10     cin >> n >> k;
11     for(int i = 0; i < n; i++){
12         for(int j = 0; j < k; j++){
13             cin >> b;
14             S[i] <<= 1;
15             S[i] |= b ^ '0';
16         }
17     }
18     for(int i = 0; i < n; i++){
19         for(int j = i + 1; j < n; j++){
20             ans = min(ans, (int)__builtin_popcount(S[i] ^ S[j]));
21         }
22     }
23     cout << ans;
24     return 0;
25 }
```

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

壓常

```
1  #include <bits/stdc++.h>
2  #pragma GCC target("popcnt")
3  #define int long long
4  using namespace std;
5  array<bitset<3000>, 3000> G;
6  signed main(){
7      cin.tie(0), cout.tie(0), ios::sync_with_stdio(0);
8      int n, ans = 0, tmp;
9      cin >> n;
10     for(int i = 0; i < n; i++){
11         cin >> G[i];
12     }
13     for(int i = 0; i < n; i++){
14         for(int j = i + 1; j < n; j++){
15             tmp = (G[i] & G[j]).count();
16             ans += tmp * (tmp - 1);
17         }
18     }
19     cout << ans / 2;
20     return 0;
21 }
```

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

壓常 Topological Sort

```
1  #include <bits/stdc++.h>
2  #define pb push_back
3  #pragma target("popcnt")
4  using namespace std;
5  array<int, 50004> in;
6  array<bitset<50004>, 50004> dp;
7  array<vector<int>, 50004> G;
8  stack<int> ord;
9  void topu(int n){
10     int u;
11     queue<int> Q;
12     for(int i = 1; i <= n; i++){
13         if(!in[i]) Q.push(i);
14     }
15     while(!Q.empty()){
16         u = Q.front();
17         Q.pop();
18         ord.push(u);
19         for(int v : G[u]){
20             in[v]--;
21             if(!in[v]) Q.push(v);
22         }
23     }
24     while(!ord.empty()){
25         u = ord.top();
26         ord.pop();
27         dp[u][u] = 1;
28         for(int v : G[u]) dp[u] |= dp[v];
29     }
30 }
31 signed main(){
32     cin.tie(0), cout.tie(0), ios::sync_with_stdio(0);
33     int n, m, a, b;
34     cin >> n >> m;
35     while(m--){
36         cin >> a >> b;
37         G[a].pb(b);
38         in[b]++;
39     }
40     topu(n);
41     for(int i = 1; i <= n; i++){
42         cout << dp[i].count() << " ";
43     }
44     return 0;
45 }
```

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

壓常 Topological Sort SCC


```
1  #include <bits/stdc++.h>
2  #define pb push_back
3  #pragma target("popcnt")
4  using namespace std;
5  int cnt = 0;
6  bitset<50004> vis;
7  array<int, 50004> in, scc;
8  array<vector<int>, 50004> G, R, S;
9  array<bitset<50004>, 50004> dp;
10 stack<int> out, ord;
11 void bfs(int u){
12     if(vis[u]) return;
13     vis[u] = 1;
14     for(int v : R[u]) bfs(v);
15     out.push(u);
16 }
17 void dfs(int u){
18     if(scc[u]) return;
19     scc[u] = cnt;
20     for(int v : G[u]) dfs(v);
21 }
22 void topu(int n){
23     int u;
24     queue<int> Q;
25     for(int i = 1; i <= n; i++){
26         if(!in[i]) Q.push(i);
27     }
28     while(!Q.empty()){
29         u = Q.front();
30         Q.pop();
31         ord.push(u);
32         for(int v : S[u]){
33             in[v]--;
34             if(!in[v]) Q.push(v);
35         }
36     }
37     while(!ord.empty()){
38         u = ord.top();
39         ord.pop();
40         dp[u][u] = 1;
41         for(int v : S[u]) dp[u] |= dp[v];
42     }
43 }
44 signed main(){
45     int n, m, q, a, b, u;
46     cin >> n >> m >> q;
47     while(m--){
48         cin >> a >> b;
49         G[a].pb(b);
50         R[b].pb(a);
51     }
52     for(int i = 1; i <= n; i++){
53         bfs(i);
```



```
54     }
55     while(!out.empty()){
56         u = out.top();
57         out.pop();
58         if(!scc[u]) cnt++;
59         dfs(u);
60     }
61     for(int i = 1; i <= n; i++){
62         for(int v : G[i]){
63             if(scc[v] == scc[i]) continue;
64             in[scc[v]]++;
65             S[scc[i]].pb(scc[v]);
66         }
67     }
68     topu(cnt);
69     while(q--){
70         cin >> a >> b;
71         cout <<(dp[scc[a]][scc[b]]? "YES\n" : "NO\n");
72     }
73     return 0;
74 }
```

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

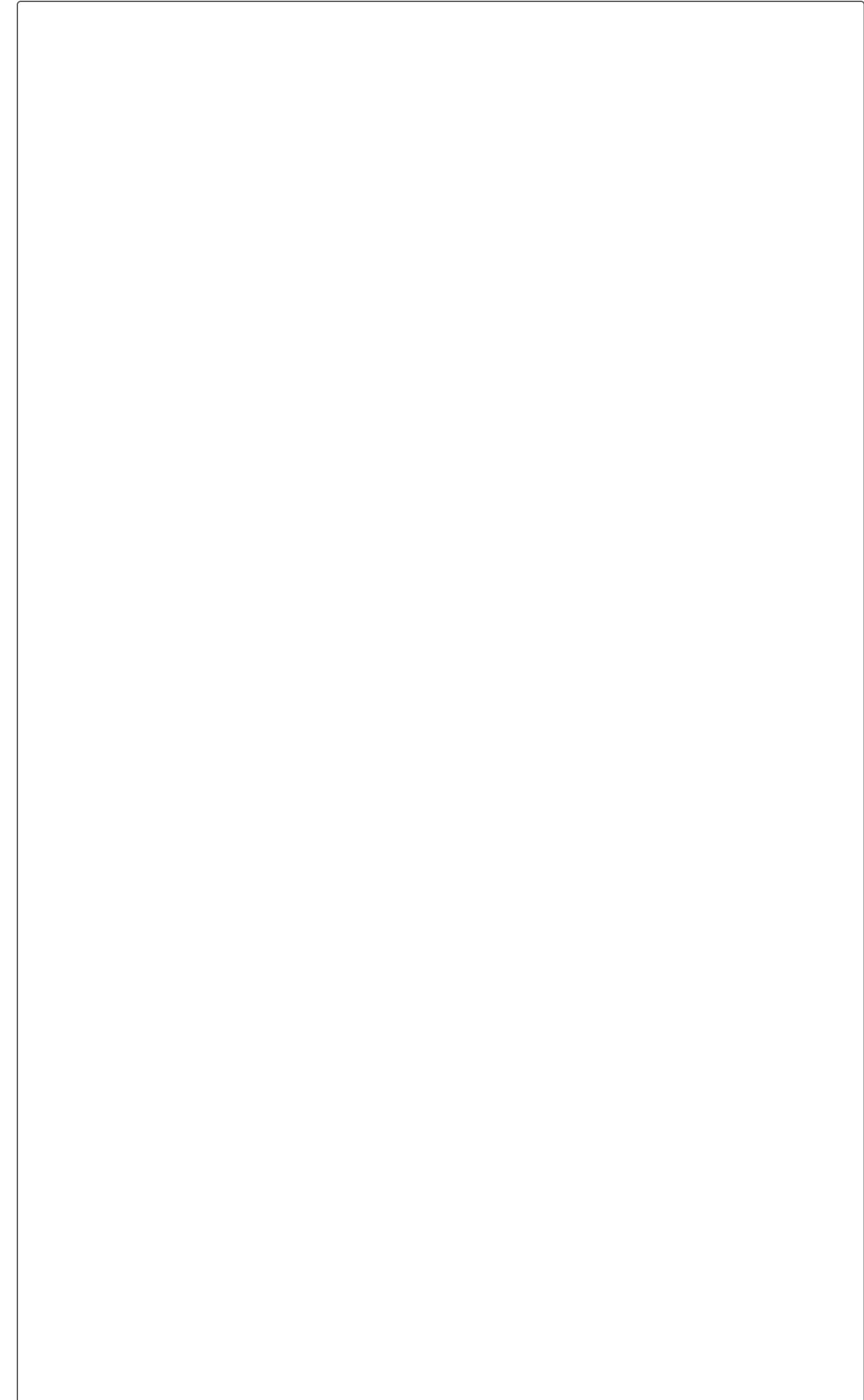
Treap


```
1  #include <bits/stdc++.h>
2  using namespace std;
3  struct treap{
4      int pri, s;
5      char x;
6      treap *lc, *rc;
7      treap(char c){
8          pri = rand();
9          lc = rc = nullptr;
10         x = c;
11         s = 1;
12     }
13     void pull(){
14         s = (lc? lc->s : 0) + (rc? rc->s : 0) + 1;
15     }
16 };
17 int size(treap *t){
18     return t? t->s : 0;
19 }
20 treap* merge(treap *a, treap *b){
21     if(!a || !b) return (!a? b : a);
22     if(a->pri > b->pri){
23         a->rc = merge(a->rc, b);
24         a->pull();
25         return a;
26     }
27     else{
28         b->lc = merge(a, b->lc);
29         b->pull();
30         return b;
31     }
32 }
33 void split(treap *t, treap *&a, treap *&b, int k){
34     if(!t){
35         a = b = nullptr;
36         return;
37     }
38     if(size(t->lc) + 1 <= k){
39         a = t;
40         split(t->rc, a->rc, b, k - size(t->lc) - 1);
41         a->pull();
42     }else{
43         b = t;
44         split(t->lc, a, b->lc, k);
45         b->pull();
46     }
47 }
48 void print(treap *t){
49     if(!t) return;
50     print(t->lc);
51     cout << t->x;
52     print(t->rc);
53 }
```

```
54 signed main(){
55     srand(time(NULL));
56     int n, q, l, r;
57     char x;
58     treap *t = nullptr, *a = nullptr, *b = nullptr, *c = nullptr;
59     cin >> n >> q;
60     for(int i = 0; i < n; i++){
61         cin >> x;
62         t = merge(t, new treap(x));
63     }
64     while(q--){
65         cin >> l >> r;
66         split(t, a, b, l - 1);
67         split(b, b, c, r - l + 1);
68         t = merge(merge(a, c), b);
69     }
70     print(t);
71     cout << "\n";
72     return 0;
73 }
```

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

Treap



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

```
54     }else{
55         b = t;
56         split(t->lc, a, b->lc, k);
57         b->pull();
58     }
59 }
60 void print(treap *t){
61     if(!t) return;
62     t->push();
63     print(t->lc);
64     cout << t->x;
65     print(t->rc);
66 }
67 signed main(){
68     srand(time(NULL));
69     int n, q, l, r;
70     char x;
71     treap *t = np, *a = np, *b = np, *c = np;
72     cin >> n >> q;
73     for(int i = 0; i < n; i++){
74         cin >> x;
75         t = merge(t, new treap(x));
76     }
77     while(q--){
78         cin >> l >> r;
79         split(t, a, b, l - 1);
80         split(b, b, c, r - l + 1);
81         b->rev ^= 1;
82         t = merge(merge(a, b), c);
83     }
84     print(t);
85     cout << "\n";
86     return 0;
87 }
```

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

Treap


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

```
54         split(t->rc, a->rc, b, k - size(t->lc) - 1);
55         a->pull();
56     }else{
57         b = t;
58         split(t->lc, a, b->lc, k);
59         b->pull();
60     }
61 }
62 signed main(){
63     int n, q, p, l, r, x;
64     treap *t = np, *a = np, *b = np, *c = np;
65     cin >> n >> q;
66     for(int i = 0; i < n; i++){
67         cin >> x;
68         t = merge(t, new treap(x));
69     }
70     while(q--){
71         cin >> p >> l >> r;
72         split(t, a, b, l - 1);
73         split(b, b, c, r - l + 1);
74         if(p == 1) b->rev ^= 1;
75         else cout << b->sum << "\n";
76         t = merge(merge(a, b), c);
77     }
78     return 0;
79 }
```

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

橋 DFS Tree

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

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

膝蓋 DFS Tree

```
1  #include <bits/stdc++.h>
2  #define pb push_back
3  using namespace std;
4  int cnt = 0;
5  array<int, 100004> P, low;
6  array<vector<int>, 100004> G;
7  vector<int> knee;
8  void dfs(int u, int pre){
9      int c = 0;
10     bool ok = 0;
11     P[u] = low[u] = ++cnt;
12     for(int v : G[u]){
13         if(v == pre) continue;
14         if(P[v]) low[u] = min(low[u], P[v]);
15         else{
16             c++;
17             dfs(v, u);
18             if(low[v] >= P[u]) ok = 1;
19             low[u] = min(low[u], low[v]);
20         }
21     }
22     if((ok && u > 1) || (u == 1 && c > 1)) knee.pb(u);
23 }
24 signed main(){
25     int n, m, a, b;
26     cin >> n >> m;
27     while(m--){
28         cin >> a >> b;
29         G[a].pb(b);
30         G[b].pb(a);
31     }
32     dfs(1, 0);
33     cout << knee.size() << "\n";
34     for(int k : knee) cout << k << " ";
35     cout << "\n";
36     return 0;
37 }
```

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

```
1  #include <bits/stdc++.h>
2  #define pb push_back
3  using namespace std;
4  const int mod = 1e9 + 7;
5  int ans = 1;
6  array<int, 100004> vis;
7  array<vector<int>, 100004>G;
8  void dfs(int u, int pre){
9      vis[u]++;
10     for(int v : G[u]){
11         if(v == pre || vis[v] > 1) continue;
12         if(vis[v] == 1) ans = ans * 2 % mod;
13         else dfs(v, u);
14     }
15     vis[u]++;
16 }
17 signed main(){
18     int n, m, a, b;
19     cin >> n >> m;
20     while(m--){
21         cin >> a >> b;
22         G[a].pb(b);
23         G[b].pb(a);
24     }
25     for(int i = 1; i <= n; i++){
26         if(!vis[i]) dfs(i, 0);
27     }
28     cout << ans << "\n";
29     return 0;
30 }
```

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

斜率優化

```
1  #include <bits/stdc++.h>
2  #define int long long
3  #define double long double
4  using namespace std;
5  struct line{
6      double a, b;
7      line(){}
8      line(double x, double y): a(x), b(y){}
9      double operator*(double x){
10         return a * x + b;
11     }
12     line operator^(line f){
13         double x, y;
14         x = (b - f.b) / (f.a - a);
15         y = f * x;
16         return line(x, y);
17     }
18 };
19 int l, r;
20 array<double, 200004> F, S, dp;
21 array<line, 200004> Q;
22 void pop(double x){
23     while(l < r && Q[l] * x > Q[l + 1] * x) l++;
24 }
25 void push(line f){
26     while(r > l){
27         auto [x, y] = f ^ Q[r - 1];
28         if(Q[r] * x >= y) r--;
29         else break;
30     }
31     Q[++r] = f;
32 }
33 int DP(int n, double x){
34     l = 0, r = -1;
35     push({x, 0});
36     for(int i = 1; i <= n; i++){
37         pop(S[i]);
38         dp[i] = Q[l] * S[i];
39         push({F[i], dp[i]});
40     }
41     return (int)dp[n];
42 }
43 signed main(){
44     int n;
45     double x;
46     cin >> n >> x;
47     for(int i = 1; i <= n; i++) cin >> S[i];
48     for(int i = 1; i <= n; i++) cin >> F[i];
49     cout << DP(n, x) << "\n";
50     return 0;
51 }
```

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

斜率優化 李超線段樹

```

1  #include <bits/stdc++.h>
2  #define int long long
3  #define mid ((l + r) >> 1)
4  #define lc (p << 1)
5  #define rc ((p << 1) | 1)
6  using namespace std;
7  struct line{
8      int a, b;
9      int operator+(int x){
10         return a * x + b;
11     }
12 };
13 const int inf = 1e6;
14 array<int, 200004> S, F, dp;
15 array<line, 400004> seg;
16 void update(int p, int l, int r, line s){
17     if(s + mid < seg[p] + mid) swap(s, seg[p]);
18     if(l == r) return;
19     if(s.a > seg[p].a) update(lc, l, mid, s);
20     else update(rc, mid + 1, r, s);
21 }
22 int query(int p, int l, int r, int x){
23     if(l == r) return seg[p] + x;
24     if(x <= mid) return min(seg[p] + x, query(lc, l, mid, x));
25     else return min(seg[p] + x, query(rc, mid + 1, r, x));
26 }
27 int DP(int n){
28     update(1, 1, inf, {F[0], 0});
29     for(int i = 1; i <= n; i++){
30         dp[i] = query(1, 1, inf, S[i]);
31         update(1, 1, inf, {F[i], dp[i]});
32     }
33     return dp[n];
34 }
35 signed main(){
36     int n, x;
37     cin >> n >> x;
38     for(int i = 1; i < 400004; i++) seg[i] = {inf, inf};
39     for(int i = 1; i <= n; i++) cin >> S[i];
40     for(int i = 1; i <= n; i++) cin >> F[i];
41     F[0] = x;
42     cout << DP(n) << "\n";
43     return 0;
44 }
```

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

分治優化

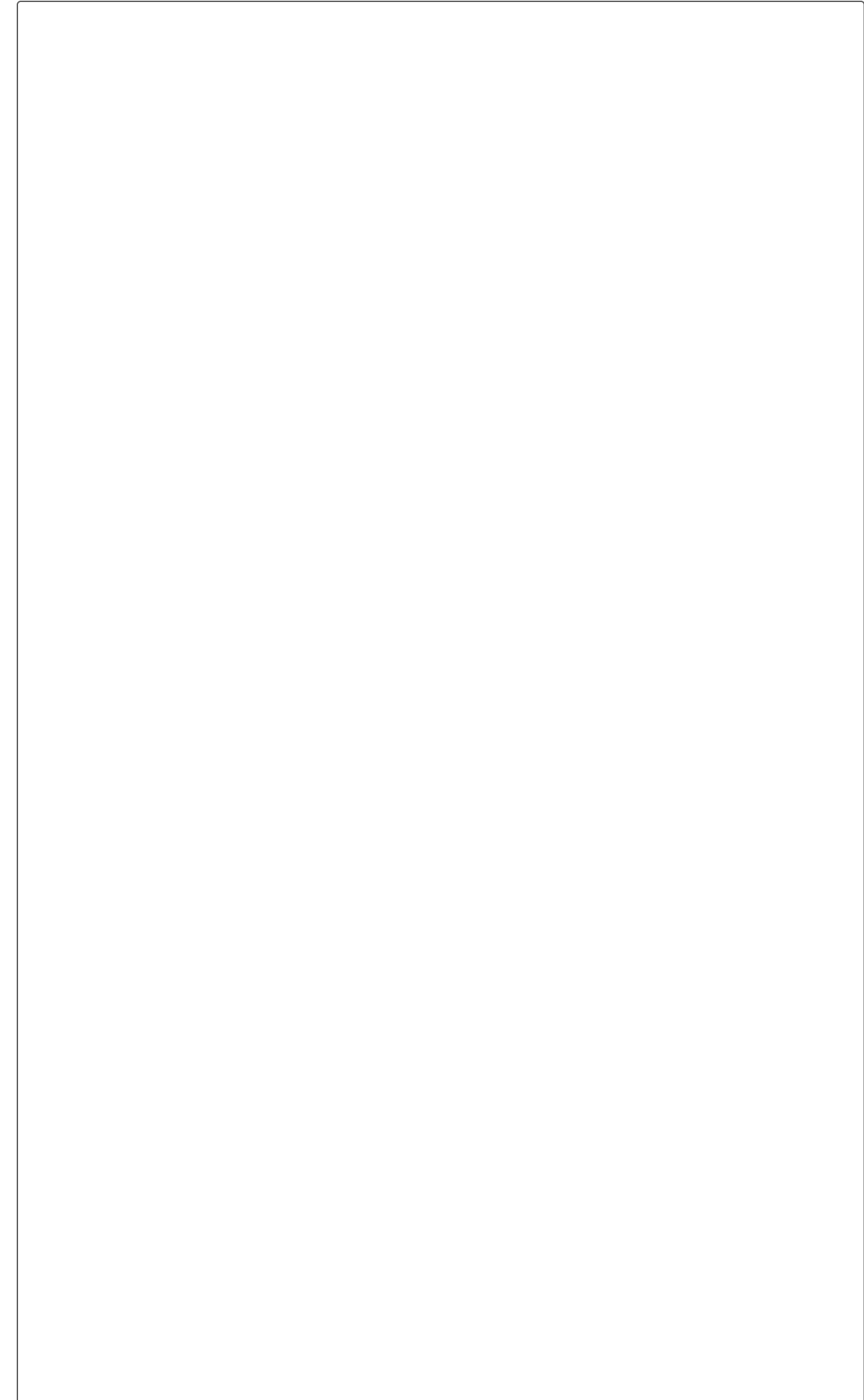
```

1  #include <bits/stdc++.h>
2  #define int long long
3  using namespace std;
4  array<int, 3004> X;
5  array<array<int, 3004>, 3004> dp;
6  int cost(int l, int r){
7      return (X[r] - X[l]) * (X[r] - X[l]);
8  }
9  void div(int ql, int qr, int l, int r, int k){
10     int t, qm = (ql + qr) >> 1;
11     dp[k][qm] = 1e18;
12     for(int i = l; i < min(r + 1, qm); i++){
13         if(dp[k - 1][i] + cost(i, qm) < dp[k][qm]){
14             t = i;
15             dp[k][qm] = dp[k - 1][i] + cost(i, qm);
16         }
17     }
18     if(ql == qr) return;
19     div(ql, qm, l, t, k);
20     div(qm + 1, qr, t, r, k);
21 }
22 int DP(int n, int k){
23     for(int i = 1; i <= n; i++){
24         dp[1][i] = X[i] * X[i];
25     }
26     for(int i = 2; i <= k; i++){
27         div(i, n, i - 1, n, i);
28     }
29     return dp[k][n];
30 }
31 signed main(){
32     int n, k;
33     cin >> n >> k;
34     for(int i = 1; i <= n; i++){
35         cin >> X[i];
36         X[i] += X[i - 1];
37     }
38     cout << DP(n, k) << "\n";
39     return 0;
40 }

```

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

分治優化



```

1  #include <bits/stdc++.h>
2  #define int long long
3  using namespace std;
4  array<int, 3004> C;
5  array<array<int, 3004>, 3004> dis, cst, turn, dp;
6  void DIS(int n){
7      for(int i = 1; i <= n; i++){
8          for(int j = i - 1; j > 0; j--){
9              dis[i][j] = (i - j) * C[j] + dis[i][j + 1];
10         }
11         for(int j = i + 1; j <= n; j++){
12             dis[i][j] = (j - i) * C[j] + dis[i][j - 1];
13         }
14     }
15     for(int i = 1; i <= n; i++){
16         cst[i][i] = 0;
17         turn[i][i] = i;
18     }
19     for(int k = 1; k < n; k++){
20         for(int i = 1, j = i + k; j <= n; i++, j++){
21             cst[i][j] = 1e18;
22             for(int t = turn[i][j - 1]; t <= turn[i + 1][j]; t++){
23                 if(dis[t][i] + dis[t][j] < cst[i][j]){
24                     cst[i][j] = dis[t][i] + dis[t][j];
25                     turn[i][j] = t;
26                 }
27             }
28         }
29     }
30 }
31 void div(int ql, int qr, int l, int r, int k){
32     int t, qm = (ql + qr) >> 1;
33     dp[k][qm] = 1e18;
34     for(int i = l; i < min(r + 1, qm); i++){
35         if(dp[k][qm] > dp[k - 1][i] + cst[i + 1][qm]){
36             dp[k][qm] = dp[k - 1][i] + cst[i + 1][qm];
37             t = i;
38         }
39     }
40     if(ql == qr) return;
41     div(ql, qm, l, t, k);
42     div(qm + 1, qr, t, r, k);
43 }
44 int DP(int n, int k){
45     for(int i = 1; i <= n; i++){
46         dp[1][i] = cst[1][i];
47     }
48     for(int i = 2; i <= k; i++){
49         div(i, n, i - 1, n, i);
50     }
51     return dp[k][n];
52 }
53 signed main(){

```

```

54     int n, k;
55     cin >> n >> k;
56     for(int i = 1; i <= n; i++) cin >> C[i];
57     DIS(n);
58     cout << DP(n, k) << "\n";
59     return 0;
60 }

```

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

Knuth 優化

```

1  #include <bits/stdc++.h>
2  #define int long long
3  using namespace std;
4  array<int, 5004> X;
5  array<array<int, 5004>, 5004> dp, turn;
6  int DP(int n){
7      for(int i = 1; i <= n; i++){
8          dp[i][i] = 0;
9          turn[i][i] = i;
10     }
11     for(int k = 1; k < n; k++){
12         for(int i = 1, j = i + k; j <= n; i++, j++){
13             dp[i][j] = 1e18;
14             for(int t = turn[i][j - 1]; t <= turn[i + 1][j]; t++){
15                 if(dp[i][t] + dp[t + 1][j] < dp[i][j]){
16                     turn[i][j] = t;
17                     dp[i][j] = dp[i][t] + dp[t + 1][j];
18                 }
19             }
20             dp[i][j] += X[j] - X[i - 1];
21         }
22     }
23     return dp[1][n];
24 }
25 signed main(){
26     int n;
27     cin >> n;
28     for(int i = 1; i <= n; i++){
29         cin >> X[i];
30         X[i] += X[i - 1];
31     }
32     cout << DP(n) << "\n";
33     return 0;
34 }

```

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

FFT


```

1  #include <bits/stdc++.h>
2  #define int long long
3  #define cmp complex<double>
4  #define r real
5  #define i imag
6  using namespace std;
7  const int N = 1 << 19;
8  const double pi = 3.14159265358979323846264;
9  array<cmp, 1 << 19> A, B, C, X;
10 cmp ei(double d){
11     return {cos(d), sin(d)};
12 }
13 void FFT(array<cmp, 1 << 19> &F){
14     int n;
15     cmp 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 k, n, m, x;
38     cin >> k >> n >> m;
39     for(int i = 0; i < n; i++){
40         cin >> x;
41         A[x] += 1;
42     }
43     for(int i = 0; i < m; i++){
44         cin >> x;
45         B[x] += 1;
46     }
47     for(int i = 0; i < N; i++){
48         X[i] = ei(2 * pi * i / N);
49     }
50     FFT(A);
51     FFT(B);
52     for(int i = 0; i < N; i++){
53         C[i] = A[i] * B[i];

```

```
54         X[i] = conj(X[i]);
55     }
56     FFT(C);
57     for(int i = 2; i <= k << 1; i++){
58         cout << rnd(C[i].r() / (double)N) << " ";
59     }
60     cout << "\n";
61     return 0;
62 }
```

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

FFT


```
1  #include <bits/stdc++.h>
2  #define cmp complex<double>
3  #define r real
4  #define i imag
5  #define int long long
6  using namespace std;
7  const int N = 1 << 19;
8  const double pi = 3.141592653589793238462643383279502884;
9  array<cmp, 1 << 19> A, B, C, X;
10 cmp ei(double d){
11     return {cos(d), sin(d)};
12 }
13 void FFT(array<cmp, 1 << 19> &F){
14     int n;
15     cmp 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;
38     string S;
39     cin >> S;
40     n = S.size();
41     for(int i = 0; i < n; i++){
42         A[i] = S[i] ^ '0';
43         B[n - i - 1] = A[i];
44     }
45     for(int i = 0; i < N; i++){
46         X[i] = ei(2 * pi * i / N);
47     }
48     FFT(A);
49     FFT(B);
50     for(int i = 0; i < N; i++){
51         C[i] = A[i] * B[i];
52         X[i] = conj(X[i]);
53     }
```

```
54 FFT(C);  
55 for(int i = n; i < (n << 1) - 1; i++){  
56     cout << rnd(C[i].r() / (double)N) << " ";  
57 }  
58 cout << "\n";  
59 return 0;  
60 }
```

Signal Processing <https://cses.fi/problemset/task/2113>

FFT


```
1  #include <bits/stdc++.h>
2  #define cmp complex<double>
3  #define r real
4  #define i imag
5  #define int long long
6  using namespace std;
7  const int N = 1 << 19;
8  const double pi = 3.14159265358979323846264338327950288419716939937
9  array<cmp, 1 << 19> A, B, C, X;
10 cmp ei(double d){
11     return {cos(d), sin(d)};
12 }
13 void FFT(array<cmp, 1 << 19> &F){
14     int n;
15     cmp 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, m;
38     cin >> n >> m;
39     for(int i = 0; i < n; i++){
40         cin >> A[i];
41     }
42     for(int i = m - 1; i >= 0; i--){
43         cin >> B[i];
44     }
45     for(int i = 0; i < N; i++){
46         X[i] = ei(2 * pi * i / N);
47     }
48     FFT(A);
49     FFT(B);
50     for(int i = 0; i < N; i++){
51         C[i] = A[i] * B[i];
52         X[i] = conj(X[i]);
53     }
```

```
54 FFT(C);
55 for(int i = 0; i < n + m - 1; i++){
56     cout << rnd(C[i].r() / (double)N) << " ";
57 }
58 cout << "\n";
59 return 0;
60 }
```

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

LCA


```
1  #include <bits/stdc++.h>
2  #define pb push_back
3  using namespace std;
4  struct E{
5      int v, t;
6  };
7  int cnt = 0;
8  array<int, 200004> in, out, dsu;
9  array<array<int, 20>, 200004> A, T;
10 array<vector<E>, 200004> G;
11 int find(int u){
12     if(dsu[u] == u) return u;
13     return dsu[u] = find(dsu[u]);
14 }
15 void union(int a, int b){
16     int A = find(a), B = find(b);
17     dsu[A] = B;
18 }
19 void dfs(int u){
20     in[u] = ++cnt;
21     for(auto [v, t] : G[u]){
22         if(in[v]) continue;
23         dfs(v);
24         T[v][0] = t;
25         A[v][0] = u;
26     }
27     out[u] = ++cnt;
28 }
29 void dabo(int n){
30     in[0] = 0, out[0] = 1e9;
31     for(int j = 1; j < 18; j++){
32         for(int i = 1; i <= n; i++){
33             A[i][j] = A[A[i][j - 1]][j - 1];
34             T[i][j] = max(T[i][j - 1], T[A[i][j - 1]][j - 1]);
35         }
36     }
37 }
38 int query(int a, int b){
39     if(find(a) != find(b)) return -1;
40     int ans = 0;
41     for(int i = 17; i >= 0; i--){
42         if(in[A[a][i]] > in[b] || out[A[a][i]] < out[b]){
43             ans = max(ans, T[a][i]);
44             a = A[a][i];
45         }
46     }
47     if(in[a] > in[b] || out[a] < out[b]){
48         ans = max(ans, T[a][0]);
49         a = A[a][0];
50     }
51     for(int i = 17; i >= 0; i--){
52         if(in[A[b][i]] > in[a] || out[A[b][i]] < out[a]){
53             ans = max(ans, T[b][i]);
```

```
54         b = A[b][i];
55     }
56 }
57 if(in[b] > in[a] || out[b] < out[a]){
58     ans = max(ans, T[b][0]);
59     b = A[b][0];
60 }
61 return ans;
62 }
63 signed main(){
64     int n, m, q, a, b;
65     cin >> n >> m >> q;
66     for(int i = 1; i <= n; i++) dsu[i] = i;
67     for(int i = 1; i <= m; i++){
68         cin >> a >> b;
69         if(find(a) == find(b)) continue;
70         G[a].pb({b, i});
71         G[b].pb({a, i});
72         union(a, b);
73     }
74     for(int i = 1; i <= n; i++){
75         if(!in[i]) dfs(i);
76     }
77     dabo(n);
78     while(q--){
79         cin >> a >> b;
80         cout << query(a, b) << "\n";
81     }
82     return 0;
83 }
```

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

Segment Tree


```
1  #include <bits/stdc++.h>
2  #define int long long
3  #define pb push_back
4  #define mid ((l + r) >> 1)
5  using namespace std;
6  struct edge{
7      int u, v, l, r;
8  };
9  int n, com;
10 array<int, 100004> DSU;
11 vector<pair<int, int>> tmp;
12 stack<vector<pair<int, int>>> chg;
13 int ehash(int u, int v){
14     return u * 100001 + v;
15 }
16 pair<int, int> dehash(int x){
17     return {x / 100001, x % 100001};
18 }
19 int find(int u){
20     tmp.pb({u, DSU[u]});
21     if(DSU[u] == u){
22         chg.push(tmp);
23         tmp.clear();
24         return u;
25     }
26     return DSU[u] = find(DSU[u]);
27 }
28 void union(int u, int v){
29     int U = find(u), V = find(v);
30     if(U == V) return;
31     DSU[V] = U;
32     com--;
33 }
34 void roll(){
35     tmp = chg.top();
36     chg.pop();
37     for(auto [u, d] : tmp){
38         DSU[u] = d;
39     }
40     tmp.clear();
41 }
42 void run(int l, int r, vector<edge> &E){
43     int c = com;
44     vector<edge> D;
45     for(edge e : E){
46         if(e.l > r || e.r < l) continue;
47         if(e.l <= l && e.r >= r) union(e.u, e.v);
48         else D.pb(e);
49     }
50     if(l == r) cout << com << " ";
51     else{
52         run(l, mid, D);
53         run(mid + 1, r, D);
```

```
54     }
55     for(edge e : E){
56         if(e.l <= l && e.r >= r) roll(), roll();
57     }
58     com = c;
59 }
60 signed main(){
61     int m, q, t, a, b, p = 0;
62     vector<edge> E;
63     map<int, int> M;
64     cin >> n >> m >> q;
65     com = n;
66     for(int i = 1; i <= n; i++) DSU[i] = i;
67     while(m--){
68         cin >> a >> b;
69         if(a > b) swap(a, b);
70         M[ehash(a, b)] = 0;
71     }
72     while(q--){
73         cin >> t >> a >> b;
74         if(a > b) swap(a, b);
75         p++;
76         if(t == 1) M[ehash(a, b)] = p;
77         else{
78             E.pb({a, b, M[ehash(a, b)], p - 1});
79             M.erase(ehash(a, b));
80         }
81     }
82     for(auto [e, s] : M){
83         auto [u, v] = dehash(e);
84         E.pb({u, v, s, p});
85     }
86     run(0, p, E);
87     return 0;
88 }
```

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

Min Cost Max Flow


```
1  #include <bits/stdc++.h>
2  #define pb push_back
3  using namespace std;
4  struct pipe{
5      int u, v, f, c;
6  };
7  int cnt = 0;
8  bitset<504> vis, in;
9  array<int, 504> cost, pre;
10 array<vector<int>, 504> G;
11 array<pipe, 2004> E;
12 void add(int u, int v, int f, int c){
13     G[u].pb(cnt);
14     E[cnt++] = {u, v, f, c};
15     G[v].pb(cnt);
16     E[cnt++] = {v, u, 0, -c};
17 }
18 void run(int u){
19     if(u == 1) return;
20     pipe &e = E[pre[u]], &b = E[pre[u] ^ 1];
21     e.f++;
22     b.f--;
23     run(e.v);
24 }
25 void SPFA(int s){
26     int u, v;
27     queue<int> Q;
28     cost[s] = 0;
29     Q.push(s);
30     while(!Q.empty()){
31         u = Q.front();
32         Q.pop();
33         vis[u] = 1;
34         in[u] = 0;
35         for(int i : G[u]){
36             if(!E[i].f) continue;
37             v = E[i].v;
38             if(cost[u] + E[i].c < cost[v]){
39                 pre[v] = i ^ 1;
40                 cost[v] = cost[u] + E[i].c;
41                 if(!in[v]){
42                     in[v] = 1;
43                     Q.push(v);
44                 }
45             }
46         }
47     }
48 }
49 int flow(int n, int k){
50     int ans = 0;
51     while(k--){
52         for(int i = 1; i <= n; i++){
53             cost[i] = 1e9;
```

```
54         vis[i] = 0;
55     }
56     SPFA(1);
57     if(!vis[n]) return -1;
58     run(n);
59     ans += cost[n];
60 }
61 return ans;
62 }
63 signed main(){
64     int n, m, k, a, b, f, c;
65     cin >> n >> m >> k;
66     while(m--){
67         cin >> a >> b >> f >> c;
68         add(a, b, f, c);
69     }
70     cout << flow(n, k) << "\n";
71     return 0;
72 }
```

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

Min Cost Max Flow


```

1  #include <bits/stdc++.h>
2  #define pb push_back
3  using namespace std;
4  struct pipe{
5      int u, v, f, c;
6  };
7  int cnt = 0;
8  bitset<412> in;
9  array<int, 412> cost, pre;
10 array<vector<int>, 412> G;
11 array<pipe, 81204> E;
12 void add(int u, int v, int f, int c){
13     G[u].pb(cnt);
14     E[cnt++] = {u, v, f, c};
15     G[v].pb(cnt);
16     E[cnt++] = {v, u, 0, -c};
17 }
18 void run(int u){
19     if(!u) return;
20     pipe &e = E[pre[u]], &b = E[pre[u] ^ 1];
21     b.f--;
22     e.f++;
23     run(e.v);
24 }
25 void SPFA(int s){
26     int u, v;
27     queue<int> Q;
28     cost[s] = 0;
29     Q.push(s);
30     while(!Q.empty()){
31         u = Q.front();
32         Q.pop();
33         in[u] = 0;
34         for(int i : G[u]){
35             if(!E[i].f) continue;
36             v = E[i].v;
37             if(cost[u] + E[i].c < cost[v]){
38                 cost[v] = cost[u] + E[i].c;
39                 pre[v] = i ^ 1;
40                 if(!in[v]){
41                     in[v] = 1;
42                     Q.push(v);
43                 }
44             }
45         }
46     }
47 }
48 int flow(int n, int k){
49     int ans = 0;
50     while(k--){
51         for(int i = 0; i <= n; i++) cost[i] = 1e9;
52         SPFA(0);
53         run(n);

```



```
54         ans += cost[n];
55     }
56     return ans;
57 }
58 signed main(){
59     int n, c;
60     cin >> n;
61     for(int i = 1; i <= n; i++){
62         add(0, i, 1, 0);
63         add(200 + i, 404, 1, 0);
64     }
65     for(int i = 1; i <= n; i++){
66         for(int j = 1; j <= n; j++){
67             cin >> c;
68             add(i, 200 + j, 1, c);
69         }
70     }
71     cout << flow(404, n) << "\n";
72     for(int i = 1; i <= n; i++){
73         cout << i << " ";
74         for(int j : G[i]){
75             if(!E[j].f) cout << E[j].v - 200 << "\n";
76         }
77     }
78     return 0;
79 }
```

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

Min Cost Max Flow


```
1  #include <bits/stdc++.h>
2  #define pb push_back
3  using namespace std;
4  struct pipe{
5      int u, v, f, c;
6  };
7  int cnt = 0;
8  bitset<504> vis, in;
9  array<int, 504> cost, pre;
10 array<vector<int>, 504> G;
11 array<pipe, 2004> E;
12 vector<int> R;
13 void add(int u, int v, int f, int c){
14     G[u].pb(cnt);
15     E[cnt++] = {u, v, f, c};
16     G[v].pb(cnt);
17     E[cnt++] = {v, u, 0, -c};
18 }
19 void run(int u){
20     if(u == 1) return;
21     pipe &e = E[pre[u]], &b = E[pre[u] ^ 1];
22     b.f--;
23     e.f++;
24     run(e.v);
25 }
26 void SPFA(int s){
27     int u, v;
28     queue<int> Q;
29     cost[s] = 0;
30     Q.push(s);
31     while(!Q.empty()){
32         u = Q.front();
33         Q.pop();
34         vis[u] = 1;
35         in[u] = 0;
36         for(int i : G[u]){
37             if(!E[i].f) continue;
38             v = E[i].v;
39             if(cost[u] + E[i].c < cost[v]){
40                 cost[v] = cost[u] + E[i].c;
41                 pre[v] = i ^ 1;
42                 if(!in[v]){
43                     in[v] = 1;
44                     Q.push(v);
45                 }
46             }
47         }
48     }
49 }
50 int flow(int n, int k){
51     int ans = 0;
52     while(k--){
53         for(int i = 1; i <= n; i++){
```

```
54         cost[i] = 1e9;
55         vis[i] = 0;
56     }
57     SPFA(1);
58     if(!vis[n]) return -1;
59     run(n);
60     ans += cost[n];
61 }
62 return ans;
63 }
64 void dfs(int u){
65     R.pb(u);
66     for(int i : G[u]){
67         if(i % 2 == 0 && !E[i].f){
68             dfs(E[i].v);
69             E[i].f++;
70             return;
71         }
72     }
73 }
74 signed main(){
75     int n, m, k, a, b;
76     cin >> n >> m >> k;
77     while(m--){
78         cin >> a >> b;
79         add(a, b, 1, 1);
80     }
81     cout << flow(n, k) << "\n";
82     while(k--){
83         R.clear();
84         dfs(1);
85         cout << R.size() << "\n";
86         for(int u : R) cout << u << " ";
87         cout << "\n";
88     }
89     return 0;
90 }
```