



# CSES PLAN II

---

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

## Range Queries

---

[Static Range Sum Queries](https://cses.fi/problemset/task/1646) (<https://cses.fi/problemset/task/1646>)

前綴

```
1 #include <bits/stdc++.h>
2 #define int long long
3 using namespace std;
4 array<int, 200004> S;
5 signed main(){
6     int n, q, l, r;
7     cin >> n >> q;
8     for(int i = 1; i <= n; i++){
9         cin >> S[i];
10        S[i] += S[i - 1];
11    }
12    while(q--){
13        cin >> l >> r;
14        cout << S[r] - S[l - 1] << "\n";
15    }
16    return 0;
17 }
```

## Static Range Minimum Queries (<https://cses.fi/problemset/task/1647>)

線段樹

```

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> S;
7 void pull(int p){
CSES PLAN II S[p] = min(S[lc], HackMD(https://hackmd.io?utm\_source=view-page&utm\_medium=logo-nav));
8 }
9
10 void update(int p, int c, int x, int l, int r){
11     if(c > r || c < l) return;
12     if(l == r){
13         S[p] = x;
14         return;
15     }
16     update(lc, c, x, l, mid);
17     update(rc, c, x, mid + 1, r);
18     pull(p);
19 }
20 int query(int p, int ql, int qr, int l, int r){
21     if(ql > r || qr < l) return 1e9;
22     if(ql <= l && qr >= r) return S[p];
23     return min(query(lc, ql, qr, l, mid), query(rc, ql, qr, mid + 1,
24 });
25 signed main(){
26     int n, q, l, r, x;
27     cin >> n >> q;
28     for(int i = 1; i <= n; i++){
29         cin >> x;
30         update(1, i, x, 1, n);
31     }
32     while(q--){
33         cin >> l >> r;
34         cout << query(1, l, r, 1, n) << "\n";
35     }
36     return 0;
37 }
```

## Dynamic Range Sum Queries (<https://cses.fi/problemset/task/1648>)

BIT

```
1 #include <bits/stdc++.h>
2 #define int long long
3 using namespace std;
4 array<int, 200004> BIT, S;
5 void update(int p, int x){
6     for(; p < 200004; p += p & -p){
7         BIT[p] += x;
8     }
9 }
10 int query(int p){
11     int sum = 0;
12     for(; p > 0; p -= p & -p){
13         sum += BIT[p];
14     }
15     return sum;
16 }
17 signed main(){
18     int n, q, t, l, r;
19     cin >> n >> q;
20     for(int i = 1; i <= n; i++){
21         cin >> S[i];
22         update(i, S[i]);
23     }
24     while(q--){
25         cin >> t >> l >> r;
26         if(t == 1){
27             update(l, r - S[l]);
28             S[l] = r;
29         }else{
30             cout << query(r) - query(l - 1) << "\n";
31         }
32     }
33     return 0;
34 }
```

## Dynamic Range Minimum Queries (<https://cses.fi/problemset/task/1649>)

線段樹

```
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> S;
7 void pull(int p){
8     S[p] = min(S[lc], S[rc]);
9 }
10 void update(int p, int c, int x, int l, int r){
11     if(c < l || c > r) return;
12     if(l == r){
13         S[p] = x;
14         return;
15     }
16     update(lc, c, x, l, mid);
17     update(rc, c, x, mid + 1, r);
18     pull(p);
19 }
20 int query(int p, int ql, int qr, int l, int r){
21     if(l > qr || r < ql) return 1e9;
22     if(ql <= l && qr >= r) return S[p];
23     return min(query(lc, ql, qr, l, mid), query(rc, ql, qr, mid + 1,
24 });
25 signed main(){
26     int n, q, t, l, r;
27     cin >> n >> q;
28     for(int i = 1; i <= n; i++){
29         cin >> t;
30         update(1, i, t, 1, n);
31     }
32     while(q--){
33         cin >> t >> l >> r;
34         if(t == 1){
35             update(1, l, r, 1, n);
36         }else{
37             cout << query(1, l, r, 1, n) << "\n";
38         }
39     }
40     return 0;
41 }
```

## Range Xor Queries (<https://cses.fi/problemset/task/1650>)

前缀

```
1 #include <bits/stdc++.h>
2 using namespace std;
3 array<int, 200004> S;
4 signed main(){
5     int n, q, l, r;
6     cin >> n >> q;
7     for(int i = 1; i <= n; i++){
8         cin >> S[i];
9         S[i] ^= S[i - 1];
10    }
11    while(q--){
12        cin >>l >> r;
13        cout << (S[r] ^ S[l - 1]) << "\n";
14    }
15    return 0;
16 }
```

## Range Update Queries (<https://cses.fi/problemset/task/1651>)

BIT

```
1 #include <bits/stdc++.h>
2 #define int long long
3 using namespace std;
4 array<int, 200004> BIT;
5 void update(int p, int x){
6     for(; p < 200004; p += p & -p){
7         BIT[p] += x;
8     }
9 }
10 int query(int p){
11     int sum = 0;
12     for(; p > 0; p -= p & -p){
13         sum += BIT[p];
14     }
15     return sum;
16 }
17 signed main(){
18     int n, q, t, l, r, x;
19     cin >> n >> q;
20     for(int i = 1; i <= n; i++){
21         cin >> x;
22         update(i, x - query(i - 1));
23     }
24     while(q--){
25         cin >> t;
26         if(t == 1){
27             cin >> l >> r >> x;
28             update(l, x);
29             update(r + 1, -x);
30         }else{
31             cin >> x;
32             cout << query(x) << "\n";
33         }
34     }
35     return 0;
36 }
```

## Forest Queries (<https://cses.fi/problemset/task/1652>)

前綴

```
1 #include <bits/stdc++.h>
2 using namespace std;
3 array<array<int, 1004>, 1004> F;
4 signed main(){
5     int n, q, x1, x2, y1, y2;
6     char f;
7     cin >> n >> q;
8     for(int i = 1; i <= n; i++){
9         for(int j = 1; j <= n; j++){
10            cin >> f;
11            if(f == '*') F[i][j]++;
12            F[i][j] += F[i - 1][j] + F[i][j - 1] - F[i - 1][j - 1];
13        }
14    }
15    while(q--){
16        cin >> x1 >> y1 >> x2 >> y2;
17        cout << F[x2][y2] - F[x2][y1 - 1] - F[x1 - 1][y2] + F[x1 - 1][y1 - 1];
18    }
19    return 0;
20 }
```

## Hotel Queries (<https://cses.fi/problemset/task/1143>)

線段樹 二分搜

```
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> S;
7 void pull(int p){
8     S[p] = max(S[lc], S[rc]);
9 }
10 void update(int p, int c, int x, int l, int r){
11     if(c < l || c > r) return;
12     if(l == r){
13         S[p] += x;
14         return;
15     }
16     update(lc, c, x, l, mid);
17     update(rc, c, x, mid + 1, r);
18     pull(p);
19 }
20 int query(int p, int x, int l, int r){
21     if(S[p] < x) return 0;
22     if(l == r) return l;
23     if(S[lc] >= x) return query(lc, x, l, mid);
24     else return query(rc, x, mid + 1, r);
25 }
26 signed main(){
27     int n, m, h, r, p;
28     cin >> n >> m;
29     for(int i = 1; i <= n; i++){
30         cin >> h;
31         update(1, i, h, 1, n);
32     }
33     while(m--){
34         cin >> r;
35         p = query(1, r, 1, n);
36         cout << p << " ";
37         if(p) update(1, p, -r, 1, n);
38     }
39     return 0;
40 }
```

## List Removals (<https://cses.fi/problemset/task/1749>)

BIT 二分搜

```
1 #include <bits/stdc++.h>
2 using namespace std;
3 array<int, 200004> BIT, S;
4 void update(int p, int x){
5     for(; p < 200004; p += p & -p){
6         BIT[p] += x;
7     }
8 }
9 int find(int x){
10     int sum = 0, p = 0;
11     for(int i = 17; i >= 0; i--){
12         if(p + (1 << i) < 200004 && sum + BIT[p + (1 << i)] < x){
13             p += (1 << i);
14             sum += BIT[p];
15         }
16     }
17     return p + 1;
18 }
19 signed main(){
20     int n, p;
21     cin >> n;
22     for(int i = 1; i <= n; i++){
23         cin >> S[i];
24         update(i, 1);
25     }
26     while(n--){
27         cin >> p;
28         cout << S[find(p)] << " ";
29         update(find(p), -1);
30     }
31     return 0;
32 }
```

## Salary Queries (<https://cses.fi/problemset/task/1144>)

動態開點線段樹



```
1 #include <bits/stdc++.h>
2 #define mid ((l + r) >> 1)
3 #define pb push_back
4 using namespace std;
5 struct node{
6     int val, lc, rc;
7 };
8 array<int, 200004> P;
9 vector<node> S;
10 void pull(int p){
11     S[p].val = S[S[p].lc].val + S[S[p].rc].val;
12 }
13 void update(int p, int c, int x, int l, int r){
14     if(c < l || c > r) return;
15     if(l == r){
16         S[p].val += x;
17         return;
18     }
19     if(!S[p].lc){
20         S[p].lc = S.size();
21         S.pb({0, 0, 0});
22     }
23     if(!S[p].rc){
24         S[p].rc = S.size();
25         S.pb({0, 0, 0});
26     }
27     update(S[p].lc, c, x, l, mid);
28     update(S[p].rc, c, x, mid + 1, r);
29     pull(p);
30 }
31 int query(int p, int ql, int qr, int l, int r){
32     if(ql > r || qr < l || !p) return 0;
33     if(ql <= l && qr >= r) return S[p].val;
34     return query(S[p].lc, ql, qr, l, mid) + query(S[p].rc, ql, qr,
35 );
36 signed main(){
37     cin.tie(0), cout.tie(0), ios::sync_with_stdio(0);
38     int n, q, l, r;
39     char t;
40     S.pb({0, 0, 0});
41     S.pb({0, 0, 0});
42     cin >> n >> q;
43     for(int i = 1; i <= n; i++){
44         cin >> P[i];
45         update(1, P[i], 1, 1, 1e9);
46     }
47     while(q--){
48         cin >> t >> l >> r;
49         if(t == '!'){
50             update(1, P[l], -1, 1, 1e9);
51             P[l] = r;
52             update(1, P[l], 1, 1, 1e9);
53         }else{
```

```
54         cout << query(1, l, r, 1, 1e9) << "\n";
55     }
56 }
57 return 0;
58 }
```

## Prefix Sum Queries (<https://cses.fi/problemset/task/2166>)

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 #define pii pair<int, int>
7 #define ff first
8 #define ss second
9 using namespace std;
10 array<int, 800004> S, P;
11 void pull(int p){
12     S[p] = S[lc] + S[rc];
13     P[p] = max(S[lc] + P[rc], P[lc]);
14 }
15 void update(int p, int c, int x, int l, int r){
16     if(c < l || c > r) return;
17     if(l == r){
18         S[p] = P[p] = x;
19         return;
20     }
21     update(lc, c, x, l, mid);
22     update(rc, c, x, mid + 1, r);
23     pull(p);
24 }
25 pii query(int p, int ql, int qr, int l, int r){
26     if(ql > r || qr < l) return {0, 0};
27     if(ql <= l && qr >= r){
28         return {P[p], S[p]};
29     }
30     pii ll, rr;
31     ll = query(lc, ql, qr, l, mid);
32     rr = query(rc, ql, qr, mid + 1, r);
33     return {max(ll.ff, ll.ss + rr.ff), ll.ss + rr.ss};
34 }
35 signed main(){
36     int n, q, t, l, r, x;
37     cin >> n >> q;
38     for(int i = 1; i <= n; i++){
39         cin >> x;
40         update(1, i, x, 1, n);
41     }
42     while(q--){
43         cin >> t >> l >> r;
44         if(t == 1){
45             update(1, l, r, 1, n);
46         }else{
47             cout << max(0ll, query(1, l, r, 1, n).ff) << "\n";
48         }
49     }
50     return 0;
51 }
```

## Pizzeria Queries (<https://cses.fi/problemset/task/2206>)

線段樹



```
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, 200004> P;
7 array<int, 800004> L, R;
8 void pull(int p){
9     L[p] = min(L[lc], L[rc]);
10    R[p] = min(R[lc], R[rc]);
11 }
12 void Lupdate(int p, int c, int x, int l, int r){
13     if(c < l || c > r) return;
14     if(l == r){
15         L[p] += x;
16         return;
17     }
18     Lupdate(lc, c, x, l, mid);
19     Lupdate(rc, c, x, mid + 1, r);
20     pull(p);
21 }
22 void Rupdate(int p, int c, int x, int l, int r){
23     if(c < l || c > r) return;
24     if(l == r){
25         R[p] += x;
26         return;
27     }
28     Rupdate(lc, c, x, l, mid);
29     Rupdate(rc, c, x, mid + 1, r);
30     pull(p);
31 }
32 int Lquery(int p, int ql, int qr, int l, int r){
33     if(ql > r || qr < l) return 2e9;
34     if(ql <= l && qr >= r) return L[p];
35     return min(Lquery(lc, ql, qr, l, mid), Lquery(rc, ql, qr, mid +
36 });
37 int Rquery(int p, int ql, int qr, int l, int r){
38     if(ql > r || qr < l) return 2e9;
39     if(ql <= l && qr >= r) return R[p];
40     return min(Rquery(lc, ql, qr, l, mid), Rquery(rc, ql, qr, mid +
41 });
42 signed main(){
43     int n, q, k, x, t;
44     cin >> n >> q;
45     for(int i = 1; i <= n; i++){
46         cin >> P[i];
47         Lupdate(1, i, i, 1, n);
48         Rupdate(1, i, n - i + 1, 1, n);
49         Lupdate(1, i, P[i], 1, n);
50         Rupdate(1, i, P[i], 1, n);
51     }
52     while(q--){
53         cin >> t >> k;
```

```
54     if(t == 1){
55         cin >> x;
56         Lupdate(1, k, x - P[k], 1, n);
57         Rupdate(1, k, x - P[k], 1, n);
58         P[k] = x;
59     }else{
60         cout << min(Lquery(1, k, n, 1, n) - k, Rquery(1, 1, k,
61             }
62         }
63     return 0;
64 }
```

## Subarray Sum Queries (<https://cses.fi/problemset/task/1190>)

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 array<int, 800004> A, P, M, S;
8 void pull(int p){
9     A[p] = A[lc] + A[rc];
10    P[p] = max(P[lc], A[lc] + P[rc]);
11    M[p] = max({S[lc] + P[rc], M[lc], M[rc]} );
12    S[p] = max(S[rc], A[rc] + S[lc]);
13 }
14 void update(int p, int c, int x, int l, int r){
15     if(c < l || c > r) return;
16     if(l == r){
17         A[p] = P[p] = M[p] = S[p] = x;
18         return;
19     }
20     update(lc, c, x, l, mid);
21     update(rc, c, x, mid + 1, r);
22     pull(p);
23 }
24 signed main(){
25     int n, m, k, x;
26     cin >> n >> m;
27     for(int i = 1; i <= n; i++){
28         cin >> x;
29         update(1, i, x, 1, n);
30     }
31     while(m--){
32         cin >> k >> x;
33         update(1, k, x, 1, n);
34         cout << max(0ll, M[1]) << "\n";
35     }
36     return 0;
37 }
```

## Distinct Values Queries (<https://cses.fi/problemset/task/1734>)

BIT

```
1 #include <bits/stdc++.h>
2 #define pb push_back
3 using namespace std;
4 struct qq{
5     int t, l, r;
6 };
7 array<int, 200004> X, BIT, ans;
8 vector<qq> Q;
9 map<int, int> P;
10 bool cmp(qq a, qq b){
11     return a.r < b.r;
12 }
13 void update(int p, int x){
14     for(; p < 200004; p += p & -p){
15         BIT[p] += x;
16     }
17 }
18 int query(int p){
19     int sum = 0;
20     for(; p > 0; p -= p & -p){
21         sum += BIT[p];
22     }
23     return sum;
24 }
25 signed main(){
26     int n, q, l, r, p = 0;
27     cin >> n >> q;
28     for(int i = 1; i <= n; i++){
29         cin >> X[i];
30     }
31     for(int i = 0; i < q; i++){
32         cin >> l >> r;
33         Q.pb({i, l, r});
34     }
35     sort(Q.begin(), Q.end(), cmp);
36     for(int i = 1; i <= n; i++){
37         if(P[X[i]]) update(P[X[i]], -1);
38         P[X[i]] = i;
39         update(i, 1);
40         while(p < Q.size() && Q[p].r == i){
41             ans[Q[p].t] = query(Q[p].r) - query(Q[p].l - 1);
42             p++;
43         }
44     }
45     for(int i = 0; i < q; i++){
46         cout << ans[i] << "\n";
47     }
48     return 0;
49 }
```

## Increasing Array Queries (<https://cses.fi/problemset/task/2416>)

懶標線段樹 二分搜



```
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 qq{
9     int l, r, t;
10 };
11 bool cmp(qq a, qq b){
12     return a.l > b.l;
13 }
14 array<int, 200004> x, ans;
15 array<int, 800004> X, S, M, tag;
16 vector<qq> Q;
17 void build(int p, int l, int r){
18     if(l == r){
19         X[p] = x[l];
20         return;
21     }
22     build(lc, l, mid);
23     build(rc, mid + 1, r);
24     X[p] = X[lc] + X[rc];
25 }
26 void pull(int p){
27     S[p] = S[lc] + S[rc];
28     M[p] = max(M[lc], M[rc]);
29 }
30 void push(int p, int l, int r){
31     if(!tag[p]) return;
32     S[lc] = (mid - l + 1) * tag[p];
33     S[rc] = (r - mid) * tag[p];
34     M[lc] = M[rc] = tag[p];
35     tag[lc] = tag[rc] = tag[p];
36     tag[p] = 0;
37 }
38 int find(int p, int v, int l, int r){
39     if(l == r) return l;
40     push(p, l, r);
41     if(M[lc] >= v) return find(lc, v, l, mid);
42     else if(M[rc] >= v) return find(rc, v, mid + 1, r);
43     else return r + 1;
44 }
45 void update(int p, int ql, int qr, int v, int l, int r){
46     if(ql > r || qr < l) return;
47     if(l != r) push(p, l, r);
48     if(ql <= l && qr >= r){
49         S[p] = (r - l + 1) * v;
50         M[p] = v;
51         tag[p] = v;
52         return;
53 }
```

```
54     update(lc, ql, qr, v, l, mid);
55     update(rc, ql, qr, v, mid + 1, r);
56     pull(p);
57 }
58 int query(int p, int ql, int qr, int l, int r){
59     if(ql > r || qr < l) return 0;
60     if(l != r) push(p, l, r);
61     if(ql <= l && qr >= r) return S[p] - X[p];
62     return query(lc, ql, qr, l, mid) + query(rc, ql, qr, mid + 1, r);
63 }
64 signed main(){
65     int n, q, l, r, p = 0;
66     cin >> n >> q;
67     for(int i = 1; i <= n; i++){
68         cin >> x[i];
69     }
70     build(1, 1, n);
71     for(int i = 0; i < q; i++){
72         cin >> l >> r;
73         Q.pb({l, r, i});
74     }
75     sort(Q.begin(), Q.end(), cmp);
76     for(int i = n; i > 0; i--){
77         update(1, i, find(1, x[i], 1, n) - 1, x[i], 1, n);
78         while(Q[p].l == i){
79             ans[Q[p].t] = query(1, Q[p].l, Q[p].r, 1, n);
80             p++;
81         }
82     }
83     for(int i = 0; i < q; i++){
84         cout << ans[i] << "\n";
85     }
86     return 0;
87 }
```

## Forest Queries II (<https://cses.fi/problemset/task/1739>)

BIT

```
1 #include <bits/stdc++.h>
2 using namespace std;
3 array<array<int, 1004>, 1004> F, BIT;
4 void update(int x, int y, int v){
5     for(int i = x; i < 1004; i += i & -i){
6         for(int j = y; j < 1004; j += j & -j){
7             BIT[i][j] += v;
8         }
9     }
10 }
11 int query(int x, int y){
12     int sum = 0;
13     for(int i = x; i > 0; i -= i & -i){
14         for(int j = y; j > 0; j -= j & -j){
15             sum += BIT[i][j];
16         }
17     }
18     return sum;
19 }
20 signed main(){
21     int n, q, x, y, x1, y1, t;
22     char f;
23     cin >> n >> q;
24     for(int i = 1; i <= n; i++){
25         for(int j = 1; j <= n; j++){
26             cin >> f;
27             if(f == '*'){
28                 update(i, j, 1);
29                 F[i][j] = 1;
30             }else{
31                 F[i][j] = 0;
32             }
33         }
34     }
35     while(q--){
36         cin >> t >> x >> y;
37         if(t == 1){
38             if(F[x][y]) update(x, y, -1);
39             else update(x, y, 1);
40             F[x][y] ^= 1;
41         }else{
42             cin >>x1 >> y1;
43             cout << query(x1, y1) - query(x1, y - 1) - query(x - 1,
44             )
45         }
46     }
47 }
```

## Range Updates and Sums (<https://cses.fi/problemset/task/1735>)

懶標線段樹



```
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 array<int, 800004> S, TAG, CHG;
8 void pull(int p){
9     S[p] = S[lc] + S[rc];
10 }
11 void push(int p, int l, int r){
12     if(CHG[p]){
13         TAG[lc] = TAG[rc] = 0;
14         CHG[lc] = CHG[rc] = CHG[p];
15         S[lc] = CHG[p] * (mid - l + 1);
16         S[rc] = CHG[p] * (r - mid);
17         CHG[p] = 0;
18     }
19     TAG[lc] += TAG[p];
20     TAG[rc] += TAG[p];
21     S[lc] += TAG[p] * (mid - l + 1);
22     S[rc] += TAG[p] * (r - mid);
23     TAG[p] = 0;
24 }
25 void update(int p, int ql, int qr, int x, int l, int r, int t){
26     if(ql > r || qr < l) return;
27     if(ql <= l && qr >= r){
28         if(t){
29             S[p] = x * (r - l + 1);
30             TAG[p] = 0;
31             CHG[p] = x;
32         }else{
33             S[p] += x * (r - l + 1);
34             TAG[p] += x;
35         }
36         return;
37     }
38     push(p, l, r);
39     update(lc, ql, qr, x, l, mid, t);
40     update(rc, ql, qr, x, mid + 1, r, t);
41     pull(p);
42 }
43 int query(int p, int ql, int qr, int l, int r){
44     if(ql > r || qr < l) return 0;
45     if(ql <= l && qr >= r) return S[p];
46     push(p, l, r);
47     return query(lc, ql, qr, l, mid) + query(rc, ql, qr, mid + 1, r);
48 }
49 signed main(){
50     int n, q, l, r, t, x;
51     cin >> n >> q;
52     for(int i = 1; i <= n; i++){
53         cin >> x;
```

```
54         update(1, i, i, x, 1, n, 0);
55     }
56     while(q--){
57         cin >> t >> l >> r;
58         if(t == 1){
59             cin >> x;
60             update(1, l, r, x, 1, n, 0);
61         }else if(t == 2){
62             cin >> x;
63             update(1, l, r, x, 1, n, 1);
64         }else{
65             cout << query(1, l, r, 1, n) << "\n";
66         }
67     }
68     return 0;
69 }
```

## Polynomial Queries (<https://cses.fi/problemset/task/1736>)

差分 懶標線段樹



```
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 array<int, 200004> T;
8 array<int, 800004> S, tag, d;
9 void pull(int p){
10     S[p] = S[lc] + S[rc];
11 }
12 void push(int p, int l, int r){
13     S[lc] += (mid - l + 1) * (tag[p] + (tag[p] + d[p] * (mid - l)));
14     S[rc] += (r - mid) * ((tag[p] + d[p] * (mid + 1 - l)) + (tag[p]
15     tag[lc] += tag[p];
16     tag[rc] += tag[p] + d[p] * (mid + 1 - l);
17     d[lc] += d[p];
18     d[rc] += d[p];
19     tag[p] = d[p] = 0;
20 }
21 void build(int p, int l, int r){
22     if(l == r){
23         S[p] = T[l];
24         return;
25     }
26     build(lc, l, mid);
27     build(rc, mid + 1, r);
28     pull(p);
29 }
30 void update(int p, int ql, int qr, int v, int l, int r){
31     if(ql > r || qr < l) return;
32     if(l != r) push(p, l, r);
33     if(ql <= l && qr >= r){
34         S[p] += (r - l + 1) * (v + (v + (r - l))) / 2ll;
35         tag[p] = v;
36         d[p] = 1;
37         return;
38     }
39     update(lc, ql, qr, v, l, mid);
40     update(rc, ql, qr, v + max(0ll, (mid + 1 - max(ql, l))), mid +
41     pull(p);
42 }
43 int query(int p, int ql, int qr, int l, int r){
44     if(ql > r || qr < l) return 0;
45     if(l != r) push(p, l, r);
46     if(ql <= l && qr >= r) return S[p];
47     return query(lc, ql, qr, l, mid) + query(rc, ql, qr, mid + 1, r
48 }
49 signed main(){
50     int n, q, t, l, r;
51     cin >> n >> q;
52     for(int i = 1; i <= n; i++){
53         cin >> T[i];
54     }
```

```
54     }
55     build(1, 1, n);
56     while(q--){
57         cin >> t >> l >> r;
58         if(t == 1){
59             update(1, l, r, 1, 1, n);
60         }else{
61             cout << query(1, l, r, 1, 1, n) << "\n";
62         }
63     }
64     return 0;
65 }
```

## Range Queries and Copies (<https://cses.fi/problemset/task/1737>)

動態開點持久化線段樹



```
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 seg{
7     int val;
8     seg *lc, *rc;
9     seg(){
10         lc = rc = nullptr;
11     }
12     void pull(){
13         val = (lc? lc->val : 0) + (rc? rc->val : 0);
14     }
15     void build(int l, int r){
16         if(l == r) return;
17         lc = new seg;
18         rc = new seg;
19         lc->build(l, mid);
20         rc->build(mid + 1, r);
21     }
22     seg* update(int c, int x, int l, int r){
23         seg *root = new seg;
24         *root = *this;
25         if(c < l || c > r) return root;
26         if(l == r){
27             root->val = x;
28             return root;
29         }
30         root->lc = lc->update(c, x, l, mid);
31         root->rc = rc->update(c, x, mid + 1, r);
32         root->pull();
33         return root;
34     }
35     int query(int ql, int qr, int l, int r){
36         if(ql > r || qr < l) return 0;
37         if(ql <= l && qr >= r) return val;
38         return lc->query(ql, qr, l, mid) + rc->query(ql, qr, mid +
39     }
40 };
41 vector<seg*> S;
42 signed main(){
43     int n, q, t, l, r, x, k;
44     cin >> n >> q;
45     S.pb(nullptr);
46     S.pb(new seg);
47     S[1]->build(1, n);
48     for(int i = 1; i <= n; i++){
49         cin >> x;
50         S[1] = S[1]->update(i, x, 1, n);
51     }
52     while(q--){
53         cin >> t >> k;
54         cout << S[t]->query(k, n) << endl;
55     }
56 }
```

```

54         if(t == 1){
55             cin >> l >> x;
56             S[k] = S[k]->update(l, x, 1, n);
57         }else if(t == 2){
58             cin >> l >> r;
59             cout << S[k]->query(l, r, 1, n) << "\n";
60         }else{
61             S.pb(S[k]);
62         }
63     }
64
65 }
```

## Tree Algorithms

---

### Subordinates (<https://cses.fi/problemset/task/1674>)

DFS

```

1 #include <bits/stdc++.h>
2 #define pb push_back
3 using namespace std;
4 array<int, 200004> ans;
5 array<vector<int>, 200004> T;
6 int dfs(int u){
7     for(int v : T[u]){
8         ans[u] += dfs(v);
9     }
10    return ans[u] + 1;
11 }
12 signed main(){
13     int n, b;
14     cin >> n;
15     for(int i = 2; i <= n; i++){
16         cin >> b;
17         T[b].pb(i);
18     }
19     dfs(1);
20     for(int i = 1; i <= n; i++){
21         cout << ans[i] << " ";
22     }
23     return 0;
24 }
```

### Tree Matching (<https://cses.fi/problemset/task/1130>)

DFS

```
1 #include <bits/stdc++.h>
2 #define pb push_back
3 using namespace std;
4 array<int, 20004> in;
5 array<vector<int>, 20004> T;
6 int dfs(int u, int pre){
7     int cnt = 0;
8     for(int v : T[u]){
9         if(v == pre) continue;
10        cnt += dfs(v, u);
11        if(!in[v] && !in[u]){
12            cnt++;
13            in[u] = 1;
14        }
15    }
16    return cnt;
17 }
18 signed main(){
19     int n, a, b;
20     cin >> n;
21     for(int i = 1; i < n; i++){
22         cin >> a >> b;
23         T[a].pb(b);
24         T[b].pb(a);
25     }
26     cout << dfs(1, 0);
27     return 0;
28 }
```

## Tree Diameter (<https://cses.fi/problemset/task/1131>)

DFS

```
1 #include <bits/stdc++.h>
2 #define pb push_back
3 using namespace std;
4 array<vector<int>, 200004> T;
5 int len = 0;
6 void comp(int &f, int &s, int k){
7     if(k >= f) swap(k, f);
8     if(k > s) swap(k, s);
9 }
10 int dfs(int u, int pre){
11     int f = 0, s = 0;
12     for(int v : T[u]){
13         if(v == pre) continue;
14         comp(f, s, dfs(v, u));
15     }
16     len = max(len, f + s);
17     return f + 1;
18 }
19 signed main(){
20     int n, a, b;
21     cin >> n;
22     for(int i = 1; i < n; i++){
23         cin >> a >> b;
24         T[a].pb(b);
25         T[b].pb(a);
26     }
27     dfs(1, 0);
28     cout << len;
29     return 0;
30 }
```

## Tree Distances I (<https://cses.fi/problemset/task/1132>)

DFS DP

```

1 #include <bits/stdc++.h>
2 #define pb push_back
3 using namespace std;
4 array<int, 20004> F, S, D;
5 array<vector<int>, 20004> T;
6 void comp(int &f, int &s, int k){
7     if(k > f) swap(k, f);
8     if(k > s) swap(k, s);
9 }
10 int dfs(int u, int pre){
11     for(int v : T[u]){
12         if(v == pre) continue;
13         comp(F[u], S[u], dfs(v, u));
14     }
15     return F[u] + 1;
16 }
17 void dsf(int u, int pre, int len){
18     D[u] = max(len, F[u]);
19     for(int v : T[u]){
20         if(v == pre) continue;
21         if(F[u] == F[v] + 1) dsf(v, u, max(S[u], len) + 1);
22         else dsf(v, u, max(F[u], len) + 1);
23     }
24 }
25 signed main(){
26     int n, a, b;
27     cin >> n;
28     for(int i = 1; i < n; i++){
29         cin >> a >> b;
30         T[a].pb(b);
31         T[b].pb(a);
32     }
33     dfs(1, 0);
34     dsf(1, 0, 0);
35     for(int i = 1; i <= n; i++){
36         cout << D[i] << " ";
37     }
38     return 0;
39 }
```

## Tree Distances II (<https://cses.fi/problemset/task/1133>)

DFS DP

```
1 #include <bits/stdc++.h>
2 #define pb push_back
3 #define int long long
4 using namespace std;
5 array<int, 200004> sub, sum, ans;
6 array<vector<int>, 200004> T;
7 int dfs(int u, int pre){
8     sub[u] = 1;
9     for(int v : T[u]){
10         if(v == pre) continue;
11         sum[u] += dfs(v, u);
12         sub[u] += sub[v];
13     }
14     return sum[u] + sub[u];
15 }
16 void dsf(int u, int pre, int sm, int sb){
17     ans[u] = sum[u] + sm;
18     for(int v : T[u]){
19         if(v == pre) continue;
20         dsf(v, u, sm + sum[u] - sum[v] - sub[v] + sb + sub[u] - sub[v]);
21     }
22 }
23 signed main(){
24     int n, a, b;
25     cin >> n;
26     for(int i = 1; i < n; i++){
27         cin >> a >> b;
28         T[a].pb(b);
29         T[b].pb(a);
30     }
31     dfs(1, 0);
32     dsf(1, 0, 0, 0);
33     for(int i = 1; i <= n; i++){
34         cout << ans[i] << " ";
35     }
36     return 0;
37 }
```

## Company Queries I (<https://cses.fi/problemset/task/1687>)

Doubling

```
1 #include <bits/stdc++.h>
2 using namespace std;
3 array<array<int, 20>, 200004> B;
4 void dabo(int n){
5     for(int j = 1; j < 20; j++){
6         for(int i = 1; i <= n; i++){
7             B[i][j] = B[B[i][j - 1]][j - 1];
8         }
9     }
10 }
11 int query(int x, int k){
12     for(int i = 19; i >= 0; i--){
13         if(k >= 1 << i){
14             x = B[x][i];
15             k -= 1 << i;
16         }
17     }
18     return x? x : -1;
19 }
20 signed main(){
21     int n, q, x, k;
22     cin >> n >> q;
23     for(int i = 2; i <= n; i++){
24         cin >> B[i][0];
25     }
26     dabo(n);
27     while(q--){
28         cin >> x >> k;
29         cout << query(x, k) << "\n";
30     }
31     return 0;
32 }
```

## Company Queries II (<https://cses.fi/problemset/task/1688>)

LCA

```
1 #include <bits/stdc++.h>
2 #define pb push_back
3 using namespace std;
4 int cnt = 0;
5 array<int, 200004> in, out;
6 array<array<int, 20>, 200004> B;
7 array<vector<int>, 200004> T;
8 void dfs(int u){
9     if(in[u]) return;
10    in[u] = ++cnt;
11    for(int v : T[u]){
12        dfs(v);
13    }
14    out[u] = ++cnt;
15 }
16 void dabo(int n){
17     in[0] = 0, out[0] = 1e9;
18     for(int j = 1; j < 20; j++){
19         for(int i = 1; i <= n; i++){
20             B[i][j] = B[B[i][j - 1]][j - 1];
21         }
22     }
23 }
24 int LCA(int a, int b){
25     for(int i = 19; i >= 0; i--){
26         if(in[B[a][i]] > in[b] || out[B[a][i]] < out[b]){
27             a = B[a][i];
28         }
29     }
30     if(in[a] <= in[b] && out[a] >= out[b]) return a;
31     return B[a][0];
32 }
33 signed main(){
34     int n, q, a, b;
35     cin >> n >> q;
36     for(int i = 2; i <= n; i++){
37         cin >> B[i][0];
38         T[B[i][0]].pb(i);
39     }
40     dfs(1);
41     dabo(n);
42     while(q--){
43         cin >> a >> b;
44         cout << LCA(a, b) << "\n";
45     }
46     return 0;
47 }
```

## Distance Queries (<https://cses.fi/problemset/task/1135>)

LCA



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

```
54     dabo(n);
55     while(q--){
56         cin >> a >> b;
57         cout << LCA(a, b) << "\n";
58     }
59     return 0;
60 }
```

## Counting Paths (<https://cses.fi/problemset/task/1136>)

LCA DP



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

```
54     P[c]--;
55     P[A[c][0]]--;
56 }
57 dsf(1, 0);
58 for(int i = 1; i <= n; i++){
59     cout << P[i] << " ";
60 }
61 return 0;
62 }
```

## Subtree Queries (<https://cses.fi/problemset/task/1137>)

樹壓平 BIT

```
1 #include <bits/stdc++.h>
2 #define int long long
3 #define pb push_back
4 using namespace std;
5 int cnt = 0;
6 array<int, 200004> BIT, L, P, V;
7 array<vector<int>, 200004> T;
8 void update(int p, int v){
9     for(; p < 200004; p += p & -p) BIT[p] += v;
10 }
11 int query(int p){
12     int sum = 0;
13     for(; p > 0; p -= p & -p) sum += BIT[p];
14     return sum;
15 }
16 int press(int u, int pre){
17     L[u] = 1e9;
18     for(int v : T[u]){
19         if(v == pre) continue;
20         L[u] = min(L[u], press(v, u));
21     }
22     P[u] = ++cnt;
23     return L[u] = min(L[u], P[u]);
24 }
25 signed main(){
26     int n, q, a, b, t, s, x;
27     cin >> n >> q;
28     for(int i = 1; i <= n; i++){
29         cin >> V[i];
30     }
31     for(int i = 1; i < n; i++){
32         cin >> a >> b;
33         T[a].pb(b);
34         T[b].pb(a);
35     }
36     press(1, 0);
37     for(int i = 1; i <= n; i++){
38         update(P[i], V[i]);
39     }
40     while(q--){
41         cin >> t >> s;
42         if(t == 1){
43             cin >> x;
44             update(P[s], x - V[s]);
45             V[s] = x;
46         }else cout << query(P[s]) - query(L[s] - 1) << "\n";
47     }
48     return 0;
49 }
```

## Path Queries (<https://cses.fi/problemset/task/1138>)

樹鍊剖分 BIT



```
1 #include <bits/stdc++.h>
2 #define int long long
3 #define pb push_back
4 using namespace std;
5 struct BIT{
6     vector<int> bit;
7     void update(int p, int v){
8         for(; p < bit.size(); p += p & -p) bit[p] += v;
9     }
10    int query(int p){
11        int sum = 0;
12        for(; p > 0; p -= p & -p) sum += bit[p];
13        return sum;
14    }
15 };
16 int cnt = 1;
17 array<int, 200004> V, H, P, S, D, pre, C;
18 array<vector<int>, 200004> T;
19 array<BIT, 200004> B;
20 int dfsiz(int u, int p, int dep){
21     int tmp, siz = 1, mx = 0;
22     D[u] = dep;
23     pre[u] = p;
24     for(int v : T[u]){
25         if(v == p) continue;
26         tmp = dfsiz(v, u, dep + 1);
27         siz += tmp;
28         if(tmp > mx){
29             mx = tmp;
30             S[u] = v;
31         }
32     }
33     return siz;
34 }
35 void cut(int u, int h, int c, int p){
36     H[u] = h;
37     C[u] = c;
38     P[u] = p;
39     if(p == 1) B[c].bit.pb(0);
40     B[c].bit.pb(0);
41     for(int v : T[u]){
42         if(v == pre[u]) continue;
43         if(v == S[u]) cut(v, h, c, p + 1);
44         else cut(v, v, ++cnt, 1);
45     }
46 }
47 int path(int s){
48     int ans = 0;
49     while(s){
50         ans += B[C[s]].query(P[s]);
51         s = pre[H[s]];
52     }
53     return ans;
```

```
54 }
55 signed main(){
56     int n, q, a, b, t, s, x;
57     cin >> n >> q;
58     for(int i = 1; i <= n; i++){
59         cin >> V[i];
60     }
61     for(int i = 1; i < n; i++){
62         cin >> a >> b;
63         T[a].pb(b);
64         T[b].pb(a);
65     }
66     dfsiz(1, 0, 1);
67     cut(1, 1, 1, 1);
68     for(int i = 1; i <= n; i++){
69         B[C[i]].update(P[i], V[i]);
70     }
71     while(q--){
72         cin >> t >> s;
73         if(t == 1){
74             cin >> x;
75             B[C[s]].update(P[s], x - V[s]);
76             V[s] = x;
77         }else cout << path(s) << "\n";
78     }
79     return 0;
80 }
```

## Path Queries II (<https://cses.fi/problemset/task/2134>)

樹鍊剖分 線段樹



```
1 #include <bits/stdc++.h>
2 #define mid ((l + r) >> 1)
3 #define pb push_back
4 using namespace std;
5 struct seg{
6     int val;
7     seg *lc, *rc;
8     seg(){val = 0; lc = rc = nullptr;}
9     void pull(){
10         val = max(lc? lc->val : 0, rc? rc->val : 0);
11     }
12     void update(int x, int v, int l, int r){
13         if(l == r){
14             val = v;
15             return;
16         }
17         if(x <= mid){
18             if(!lc) lc = new seg;
19             lc->update(x, v, l, mid);
20         }else{
21             if(!rc) rc = new seg;
22             rc->update(x, v, mid + 1, r);
23         }
24         pull();
25     }
26     int query(int ql, int qr, int l, int r){
27         if(ql > r || qr < l) return 0;
28         if(ql <= l && qr >= r) return val;
29         return max(lc? lc->query(ql, qr, l, mid) : 0, rc? rc->quer
30     }
31 };
32 int cnt = 1, n;
33 array<int, 200004> P, C, D, pre, H, M, V, Z;
34 array<vector<int>, 200004> T;
35 array<seg*, 200004> S;
36 int dfs(int u, int p, int dep){
37     int s = 1, tmp, mx = 0;
38     D[u] = dep;
39     pre[u] = p;
40     for(int v : T[u]){
41         if(v == p) continue;
42         tmp = dfs(v, u, dep + 1);
43         s += tmp;
44         if(tmp > mx){
45             mx = tmp;
46             M[u] = v;
47         }
48     }
49     return s;
50 }
51 void cut(int u, int h, int c, int p){
52     H[u] = h;
53     C[u] = c;
```

```
54     P[u] = p;
55     Z[c] = p;
56     for(int v : T[u]){
57         if(v == pre[u]) continue;
58         if(v == M[u]) cut(v, h, c, p + 1);
59         else cut(v, v, ++cnt, 1);
60     }
61 }
62 int path(int a, int b){
63     int ans = 0;
64     while(H[a] != H[b]){
65         if(D[H[a]] < D[H[b]]) swap(a, b);
66         ans = max(ans, S[C[a]]->query(1, P[a], 1, Z[C[a]]));
67         a = pre[H[a]];
68     }
69     if(D[a] < D[b]) swap(a, b);
70     return max(ans, S[C[a]]->query(P[b], P[a], 1, Z[C[a]]));
71 }
72 signed main(){
73     cin.tie(0), cout.tie(0), ios::sync_with_stdio(0);
74     int q, a, b, t, s, x;
75     cin >> n >> q;
76     for(int i = 1; i <= n; i++){
77         cin >> V[i];
78     }
79     for(int i = 1; i < n; i++){
80         cin >> a >> b;
81         T[a].pb(b);
82         T[b].pb(a);
83     }
84     dfs(1, 0, 1);
85     cut(1, 1, 1, 1);
86     for(int i = 1; i <= n; i++){
87         if(!S[C[i]]) S[C[i]] = new seg;
88         S[C[i]]->update(P[i], V[i], 1, Z[C[i]]);
89     }
90     while(q--){
91         cin >> t;
92         if(t == 1){
93             cin >> s >> x;
94             S[C[s]]->update(P[s], x, 1, Z[C[s]]);
95         }else{
96             cin >> a >> b;
97             cout << path(a, b) << "\n";
98         }
99     }
100    return 0;
101 }
```

## Distinct Colors (<https://cses.fi/problemset/task/1139>)

Set 啟發式合併

```

1 #include <bits/stdc++.h>
2 #define pb push_back
3 #define ins insert
4 using namespace std;
5 array<int, 200004> C, ans;
6 array<vector<int>, 200004> T;
7 set<int> dfs(int u, int pre){
8     set<int> S, tmp;
9     S.ins(C[u]);
10    for(int v : T[u]){
11        if(v == pre) continue;
12        tmp = dfs(v, u);
13        if(tmp.size() > S.size()) swap(S, tmp);
14        for(int it : tmp){
15            S.ins(it);
16        }
17    }
18    ans[u] = S.size();
19    return S;
20 }
21 signed main(){
22     int n, a, b;
23     cin >> n;
24     for(int i = 1; i <= n; i++){
25         cin >> C[i];
26     }
27     for(int i = 1; i < n; i++){
28         cin >> a >> b;
29         T[a].pb(b);
30         T[b].pb(a);
31     }
32     dfs(1, 0);
33     for(int i = 1; i <= n; i++){
34         cout << ans[i] << " ";
35     }
36     return 0;
37 }
```

## Finding a Centroid (<https://cses.fi/problemset/task/2079>)

DFS

```
1 #include <bits/stdc++.h>
2 #define pb push_back
3 using namespace std;
4 int n, cen;
5 array<vector<int>, 200004> T;
6 int dfs(int u, int pre){
7     int s = 1, mux = 0, tmp;
8     for(int v : T[u]){
9         if(v == pre) continue;
10        tmp = dfs(v, u);
11        s += tmp;
12        mux = max(mux, tmp);
13    }
14    if(max(mux, n - s) <= n / 2) cen = u;
15    return s;
16 }
17 signed main(){
18     int a, b;
19     cin >> n;
20     for(int i = 1; i < n; i++){
21         cin >> a >> b;
22         T[a].pb(b);
23         T[b].pb(a);
24     }
25     dfs(1, 0);
26     cout << cen;
27     return 0;
28 }
```

## Fixed-Length Paths I (<https://cses.fi/problemset/task/2080>)

重心剖分



```
1 #include <bits/stdc++.h>
2 #define int long long
3 #define pb push_back
4 using namespace std;
5 int n, k;
6 array<bool, 200004> vis;
7 array<int, 200004> S, M, cnt;
8 array<vector<int>, 200004> T, C;
9 vector<int> leaf;
10 int dfsiz(int u){
11     if(vis[u]) return 0;
12     int tmp;
13     leaf.pb(u);
14     vis[u] = 1;
15     S[u] = 1;
16     M[u] = 0;
17     for(int v : T[u]){
18         tmp = dfsiz(v);
19         M[u] = max(M[u], tmp);
20         S[u] += tmp;
21     }
22     return S[u];
23 }
24 int cut(int root){
25     leaf.clear();
26     int cen, s;
27     dfsiz(root);
28     s = leaf.size();
29     for(int u : leaf){
30         if(max(M[u], s - S[u]) <= s / 2) cen = u;
31         vis[u] = 0;
32     }
33     vis[cen] = 1;
34     for(int v : T[cen]){
35         if(!vis[v]) C[cen].pb(cut(v));
36     }
37     return cen;
38 }
39 int dfs(int u, int pre, int s){
40     if(vis[u] || s > k) return 0;
41     int sum = 0;
42     sum += cnt[k - s];
43     cnt[s]++;
44     for(int v : T[u]){
45         if(v == pre) continue;
46         sum += dfs(v, u, s + 1);
47     }
48     return sum;
49 }
50 int path(int u){
51     int ans = 0, tmp;
52     ans += dfs(u, u, 0);
53     for(int i = 0; i <= k && cnt[i]; i++) cnt[i] = 0;
```

```
54     vis[u] = 1;
55     for(int v : T[u]){
56         tmp = dfs(v, u, 1);
57         ans -= tmp;
58         for(int i = 1; i <= k && cnt[i]; i++) cnt[i] = 0;
59     }
60     for(int v : C[u]){
61         ans += path(v);
62     }
63     return ans;
64 }
65 signed main(){
66     int a, b, c;
67     cin >> n >> k;
68     for(int i = 1; i < n; i++){
69         cin >> a >> b;
70         T[a].pb(b);
71         T[b].pb(a);
72     }
73     c = cut(1);
74     for(bool &v : vis) v = 0;
75     cout << path(c);
76     return 0;
77 }
```

## Fixed-Length Paths II (<https://cses.fi/problemset/task/2081>)

重心部分 BIT



```
1 #include <bits/stdc++.h>
2 #define int long long
3 #define pb push_back
4 using namespace std;
5 int n, k1, k2, d;
6 array<bool, 200004> vis;
7 array<int, 200004> S, M, BNT;
8 array<vector<int>, 200004> T, C;
9 vector<int> leaf, see;
10 void update(int p){
11     p++;
12     for(; p <= n; p += p & -p){
13         if(!BNT[p]) see.pb(p);
14         BNT[p]++;
15     }
16 }
17 int query(int p){
18     p++;
19     if(p <= 0) return 0;
20     int sum = 0;
21     for(; p > 0; p -= p & -p) sum += BNT[p];
22     return sum;
23 }
24 int dfsiz(int u){
25     if(vis[u]) return 0;
26     int tmp;
27     leaf.pb(u);
28     vis[u] = 1;
29     S[u] = 1;
30     M[u] = 0;
31     for(int v : T[u]){
32         tmp = dfsiz(v);
33         S[u] += tmp;
34         M[u] = max(M[u], tmp);
35     }
36     return S[u];
37 }
38 int dfs(int u, int pre, int s){
39     if(vis[u] || s > k2) return 0;
40     int sum = 0;
41     sum += query(k2 - s) - query(k1 - s - 1);
42     update(s);
43     for(int v : T[u]){
44         if(v == pre) continue;
45         sum += dfs(v, u, s + 1);
46     }
47     return sum;
48 }
49 int cut(int root){
50     int cen, s, ans = 0;
51     leaf.clear();
52     dfsiz(root);
53     s = leaf.size();
```

```
54     for(int u : leaf){
55         if(max(M[u], s - S[u]) <= s / 2) cen = u;
56         vis[u] = 0;
57     }
58     ans += dfs(cen, cen, 0);
59     for(int s : see) BNT[s] = 0;
60     see.clear();
61     vis[cen] = 1;
62     for(int v : T[cen]){
63         ans -= dfs(v, cen, 1);
64         for(int s : see) BNT[s] = 0;
65         see.clear();
66     }
67     for(int v : T[cen]){
68         if(!vis[v]) ans += cut(v);
69     }
70     return ans;
71 }
72 signed main(){
73     cin.tie(0), cout.tie(0), ios::sync_with_stdio(0);
74     int a, b;
75     cin >> n >> k1 >> k2;
76     for(int i = 1; i < n; i++){
77         cin >> a >> b;
78         T[a].pb(b);
79         T[b].pb(a);
80     }
81     cout << cut(1);
82     return 0;
83 }
```

## Mathematics

---

### Josephus Queries (<https://cses.fi/problemset/task/2164>)

遞迴

```

1 #include <bits/stdc++.h>
2 using namespace std;
3 int jos(int n, int k, int f){
4     if(n == 1) return 1;
5     int kill = (n + f) / 2, nk;
6     if(k <= kill) return 2 * k - f;
7     else{
8         nk = jos(n - kill, k - kill, (n ^ f) & 1);
9         return 2 * nk - (1 ^ f);
10    }
11 }
12 signed main(){
13     int q, n, k;
14     cin >> q;
15     while(q--){
16         cin >> n >> k;
17         cout << jos(n, k, 0) << "\n";
18     }
19     return 0;
20 }
```

## Exponentiation (<https://cses.fi/problemset/task/1095>)

快速幂

```

1 #include <bits/stdc++.h>
2 #define int long long
3 using namespace std;
4 const int mod = 1e9 + 7;
5 int power(int x, int k){
6     int ans = 1;
7     for(int i = 1; i <= k; i <= 1){
8         if(k & i) ans *= x, ans %= mod;
9         x *= x, x %= mod;
10    }
11    return ans;
12 }
13 signed main(){
14     int n, a, b;
15     cin >> n;
16     while(n--){
17         cin >> a >> b;
18         cout << power(a, b) << "\n";
19     }
20     return 0;
21 }
```

## Exponentiation II (<https://cses.fi/problemset/task/1712>)

快速幂 費馬小定理

```

1 #include <bits/stdc++.h>
2 #define int long long
3 using namespace std;
4 int power(int x, int k, int p){
5     int ans = 1;
6     for(int i = 1; i <= k; i <= 1){
7         if(i & k) ans *= x, ans %= p;
8         x *= x, x %= p;
9     }
10    return ans;
11 }
12 signed main(){
13     int n, a, b, c, bc, p = 1e9 + 7;
14     cin >> n;
15     while(n--){
16         cin >> a >> b >> c;
17         bc = power(b, c, p - 1);
18         cout << power(a, bc, p) << "\n";
19     }
20     return 0;
21 }
```

## Counting Divisors (<https://cses.fi/problemset/task/1713>)

根號

```

1 #include <bits/stdc++.h>
2 using namespace std;
3 int div(int x){
4     int cnt = 0;
5     for(int i = 1; i * i <= x; i++){
6         if(x % i == 0) cnt += 2;
7         if(i * i == x) cnt--;
8     }
9     return cnt;
10 }
11 signed main(){
12     int n, x;
13     cin >> n;
14     while(n--){
15         cin >> x;
16         cout << div(x) << "\n";
17     }
18     return 0;
19 }
```

## Common Divisors (<https://cses.fi/problemset/task/1081>)

類質數篩

```
1 #include <bits/stdc++.h>
2 using namespace std;
3 array<int, 1000004> cnt;
4 int gcd(){
5     int div;
6     for(int i = 1e6; i > 0; i--){
7         div = 0;
8         for(int j = i; j <= 1e6; j += i){
9             div += cnt[j];
10        }
11        if(div > 1) return i;
12    }
13 }
14 signed main(){
15     int n, x;
16     cin >> n;
17     while(n--){
18         cin >> x;
19         cnt[x]++;
20     }
21     cout << gcd();
22     return 0;
23 }
```

## Sum of Divisors (<https://cses.fi/problemset/task/1082>)

根號 模逆元

```
1 #include <bits/stdc++.h>
2 #define int long long
3 using namespace std;
4 const int mod = 1e9 + 7, m = 5e8 + 4;
5 signed main(){
6     int n, ans = 0, l, r, gh;
7     cin >> n;
8     gh = sqrt(n);
9     for(int i = 1; i <= gh; i++){
10         l = n / i, r = n / (i + 1) + 1;
11         ans += (((((l - r + 1) % mod) * ((l + r) % mod)) % mod) *
12             ans % mod;
13     }
14     for(int i = 1; i <= n / (gh + 1); i++){
15         ans += (n / i) * i;
16         ans %= mod;
17     }
18     cout << ans;
19     return 0;
20 }
```

## Divisor Analysis (<https://cses.fi/problemset/task/2182>)

快速幂 模逆元

```
1 #include <bits/stdc++.h>
2 #define int long long
3 using namespace std;
4 const int mod = 1e9 + 7;
5 int exp(int x, int k){
6     int ans = 1;
7     for(int i = 1; i <= k; i <= 1){
8         if(i & k) ans *= x, ans %= mod;
9         x *= x, x %= mod;
10    }
11    return ans;
12 }
13 signed main(){
14     int n, x, k, num = 1, sum = 1, pro = 1, cnt = 1;
15     cin >> n;
16     for(int i = 0; i < n; i++){
17         cin >> x >> k;
18         num *= (k + 1), num %= mod;
19         sum *= (((exp(x, k + 1) - 1 + mod) % mod) * exp(x - 1, mod));
20         pro = (exp(pro, k + 1) * exp(exp(x, k * (k + 1) / 2), cnt));
21         cnt *= (k + 1), cnt %= (mod - 1);
22     }
23     cout << num << " " << sum << " " << pro;
24     return 0;
25 }
```

## Prime Multiples (<https://cses.fi/problemset/task/2185>)

位元枚舉 排容

```
1 #include <bits/stdc++.h>
2 #define int long long
3 using namespace std;
4 array<int, 20> A;
5 int mul(int n, int k){
6     int ans = 0, tmp, cnt;
7     for(int i = 1; i < 1 << k; i++){
8         tmp = n, cnt = 0;
9         for(int j = 0; j < k; j++){
10            if(i & (1 << j)){
11                tmp /= A[j];
12                cnt++;
13            }
14        }
15        if(cnt & 1) ans += tmp;
16        else ans -= tmp;
17    }
18    return ans;
19 }
20 signed main(){
21     int n, k;
22     cin >> n >> k;
23     for(int i = 0; i < k; i++){
24         cin >> A[i];
25     }
26     cout << mul(n, k);
27     return 0;
28 }
```

## Counting Coprime Pairs (<https://cses.fi/problemset/task/2417>)

因數分解 質數篩



```
1 #include <bits/stdc++.h>
2 #define int long long
3 #define pb push_back
4 using namespace std;
5 array<int, 1000004> cnt;
6 array<bool, 1000004> pri;
7 vector<int> P, C;
8 void colander(int n){
9     pri[1] = 1;
10    for(int i = 2; i <= n; i++){
11        if(pri[i]) continue;
12        for(int j = 2 * i; j <= n; j += i){
13            pri[j] = 1;
14        }
15    }
16 }
17 void dfs(int p, int x){
18     if(p >= P.size()){
19         cnt[x]++;
20         return;
21     }
22     int tmp = 1;
23     for(int i = 0; i <= C[p]; i++){
24         dfs(p + 1, x * tmp);
25         tmp *= P[p];
26     }
27 }
28 void div(int x){
29     P.clear();
30     C.clear();
31     if(!pri[x]){
32         cnt[x]++;
33         cnt[1]++;
34         return;
35     }
36     for(int i = 2; i * i <= x; i++){
37         if(x % i == 0){
38             P.pb(i);
39             C.pb(1);
40             x /= i;
41         }
42         while(x % i == 0){
43             C[C.size() - 1]++;
44             x /= i;
45         }
46     }
47     if(x > 1) P.pb(x), C.pb(1);
48     dfs(0, 1);
49 }
50 void gcd(int i){
51     cnt[i] = cnt[i] * (cnt[i] - 1) / 2;
52     for(int j = 2 * i; j <= 1e6; j += i) cnt[i] -= cnt[j];
53 }
```

```
54 signed main(){
55     cin.tie(0), cout.tie(0), ios::sync_with_stdio(0);
56     int n, x;
57     cin >> n;
58     colander(1e6);
59     for(int i = 0; i < n; i++){
60         cin >> x;
61         div(x);
62     }
63     for(int i = 1e6; i > 0; i--){
64         gcd(i);
65     }
66     cout << cnt[1];
67     return 0;
68 }
```

## Binomial Coefficients (<https://cses.fi/problemset/task/1079>)

組合

```
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 fact(int n){
7     fac[0] = 1;
8     for(int i = 1; i <= n; i++){
9         fac[i] = (fac[i - 1] * i) % 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 *= x, ans %= mod;
16         x *= x, x %= mod;
17     }
18     return ans;
19 }
20 int C(int n, int k){
21     return (((fac[n] * exp(fac[k], mod - 2)) % mod) * exp(fac[n - k], mod - 1)) % mod;
22 }
23 signed main(){
24     int n, a, b;
25     cin >> n;
26     fact(1e6);
27     while(n--){
28         cin >> a >> b;
29         cout << C(a, b) << "\n";
30     }
31     return 0;
32 }
```

## Creating Strings II (<https://cses.fi/problemset/task/1715>)

排列

```
1 #include <bits/stdc++.h>
2 #define int long long
3 using namespace std;
4 int mod = 1e9 + 7;
5 array<int, 1000004> fac;
6 array<int, 26> cnt;
7 void fact(int n){
8     fac[0] = 1;
9     for(int i = 1; i <= n; i++){
10         fac[i] = (fac[i - 1] * i) % mod;
11     }
12 }
13 int exp(int x, int k){
14     int ans = 1;
15     for(int i = 1; i <= k; i <= 1){
16         if(i & k) ans = (ans * x) % mod;
17         x = (x * x) % mod;
18     }
19     return ans;
20 }
21 signed main(){
22     int ans;
23     string S;
24     cin >> S;
25     fact(1e6);
26     for(char s : S){
27         cnt[s - 'a']++;
28     }
29     ans = fac[S.size()];
30     for(int i = 0; i < 26; i++){
31         ans = (ans * exp(fac[cnt[i]], mod - 2)) % mod;
32     }
33     cout << ans;
34     return 0;
35 }
```

## Distributing Apples (<https://cses.fi/problemset/task/1716>)

組合

```
1 #include <bits/stdc++.h>
2 #define int long long
3 using namespace std;
4 const int mod = 1e9 + 7;
5 array<int, 2000004> fac;
6 void fact(int n){
7     fac[0] = 1;
8     for(int i = 1; i <= n; i++){
9         fac[i] = (fac[i - 1] * i) % 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[k], mod - 2)) % mod) * exp(fac[n - k], mod - 1)) % mod;
22 }
23 signed main(){
24     int n, m;
25     fact(2e6);
26     cin >> n >> m;
27     cout << C(m + n - 1, n - 1);
28     return 0;
29 }
```

## Christmas Party (<https://cses.fi/problemset/task/1717>)

組合 排容

```
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 fact(int n){
7     fac[0] = 1;
8     for(int i = 1; i <= n; i++){
9         fac[i] = (fac[i - 1] * i) % 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[k], mod - 2)) % mod) * exp(fac[n - k], mod - 1)) % mod;
22 }
23 signed main(){
24     int n, ans = 0;
25     fact(1e6);
26     cin >> n;
27     for(int i = n; i >= 0; i--){
28         if((n - i) & 1) ans -= (C(n, i) * fac[i]) % mod;
29         else ans += (C(n, i) * fac[i]) % mod;
30         ans = (ans + mod) % mod;
31     }
32     cout << ans;
33     return 0;
34 }
```

## Bracket Sequences I (<https://cses.fi/problemset/task/2064>)

卡特蘭數

```
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 fact(int n){
7     fac[0] = 1;
8     for(int i = 1; i <= n; i++){
9         fac[i] = (fac[i - 1] * i) % 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[k], mod - 2)) % mod) * exp(fac[n - k], mod - 2)) % mod;
22 }
23 signed main(){
24     int n;
25     fact(1e6);
26     cin >> n;
27     if(n & 1) cout << 0;
28     else cout << (C(n, n / 2) * exp(n / 2 + 1, mod - 2)) % mod;
29     return 0;
30 }
```

## Bracket Sequences II (<https://cses.fi/problemset/task/2187>)

卡特蘭數

```

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 fact(int n){
7     fac[0] = 1;
8     for(int i = 1; i <= n; i++){
9         fac[i] = (fac[i - 1] * i) % 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 signed main(){
21     int n, l = 0, r = 0, ans;
22     string S;
23     cin >> n >> S;
24     fact(1e6);
25     if(n & 1){
26         cout << 0;
27         return 0;
28     }
29     n /= 2;
30     for(char s : S){
31         if(s == '(') l++;
32         if(s == ')') r++;
33         if(r > l){
34             cout << 0;
35             return 0;
36         }
37     }
38     if(l > n){
39         cout << 0;
40         return 0;
41     }
42     ans = (((fac[2 * n - l - r] * exp(fac[n - l], mod - 2)) % mod)
43     cout << (ans - (((fac[2 * n - l - r] * exp(fac[n - r + 1], mod
44     return 0;
45 }

```

## Counting Necklaces (<https://cses.fi/problemset/task/2209>)

Burn Side Lemma

```
1 #include <bits/stdc++.h>
2 #define int long long
3 using namespace std;
4 const int mod = 1e9 + 7;
5 int gcd(int a, int b){
6     return b? gcd(b, a % b) : a;
7 }
8 int exp(int x, int k){
9     int ans = 1;
10    for(int i = 1; i <= k; i <= 1){
11        if(i & k) ans = (ans * x) % mod;
12        x = (x * x) % mod;
13    }
14    return ans;
15 }
16 signed main(){
17     int n, m, ans = 0;
18     cin >> n >> m;
19     for(int i = 0; i < n; i++){
20         ans = (ans + exp(m, gcd(n, i))) % mod;
21     }
22     cout << (ans * exp(n, mod - 2)) % mod;
23     return 0;
24 }
```

## Counting Grids (<https://cses.fi/problemset/task/2210>)

Burn Side Lemma

```
1 #include <bits/stdc++.h>
2 #define int long long
3 using namespace std;
4 const int mod = 1e9 + 7;
5 int exp(int x, int k){
6     int ans = 1;
7     for(int i = 1; i <= k; i <= 1){
8         if(i & k) ans = (ans * x) % mod;
9         x = (x * x) % mod;
10    }
11    return ans;
12 }
13 signed main(){
14     int n, n0, n1, n2, ans = 0;
15     cin >> n;
16     if(n == 1){
17         cout << 2;
18         return 0;
19     }
20     n0 = n * n;
21     if(n & 1){
22         n1 = (n * n - 1) / 4 + 1;
23         n2 = (n * n - 1) / 2 + 1;
24     }else{
25         n1 = n * n / 4;
26         n2 = n * n / 2;
27     }
28     ans += exp(2, n0);
29     ans += exp(2, n1 + 1);
30     ans += exp(2, n2);
31     ans %= mod;
32     ans = (ans * exp(4, mod - 2)) % mod;
33     cout << ans;
34     return 0;
35 }
```

## Fibonacci Numbers (<https://cses.fi/problemset/task/1722>)

矩阵快速幂

```
1 #include <bits/stdc++.h>
2 #define int long long
3 #define matrix array<array<int, 2>, 2>
4 using namespace std;
5 const int mod = 1e9 + 7;
6 matrix K;
7 matrix mul(matrix A, matrix B){
8     matrix C;
9     for(int i = 0; i < 2; i++){
10         for(int j = 0; j < 2; j++){
11             C[i][j] = 0;
12             for(int k = 0; k < 2; k++){
13                 C[i][j] += (A[i][k] * B[k][j]) % mod;
14             }
15             C[i][j] %= mod;
16         }
17     }
18     return C;
19 }
20 int fib(int n){
21     matrix F;
22     F[0][0] = 1, F[0][1] = 1, F[1][0] = 1, F[1][1] = 0;
23     F[0][0] = 1, F[0][1] = 0, F[1][0] = 0, F[1][1] = 1;
24     for(int i = 1; i <= n; i <= 1){
25         if(i & n) F = mul(K, F);
26         K = mul(K, K);
27     }
28     return F[0][1];
29 }
30 signed main(){
31     int n;
32     cin >> n;
33     cout << fib(n);
34     return 0;
35 }
```

## Throwing Dice (<https://cses.fi/problemset/task/1096>)

矩阵快速幂

```

1 #include <bits/stdc++.h>
2 #define int long long
3 #define matrix array<array<int, 6>, 6>
4 using namespace std;
5 const int mod = 1e9 + 7;
6 matrix K;
7 matrix mul(matrix A, matrix B){
8     matrix C;
9     for(int i = 0; i < 6; i++){
10         for(int j = 0; j < 6; j++){
11             C[i][j] = 0;
12             for(int k = 0; k < 6; k++){
13                 C[i][j] += (A[i][k] * B[k][j]) % mod;
14             }
15             C[i][j] %= mod;
16         }
17     }
18     return C;
19 }
20 int dice(int n){
21     matrix D;
22     D[0][0] = 1, D[0][1] = 0, D[0][2] = 0, D[0][3] = 0, D[0][4] = 0
23     D[1][0] = 0, D[1][1] = 1, D[1][2] = 0, D[1][3] = 0, D[1][4] = 0
24     D[2][0] = 0, D[2][1] = 0, D[2][2] = 1, D[2][3] = 0, D[2][4] = 0
25     D[3][0] = 0, D[3][1] = 0, D[3][2] = 0, D[3][3] = 1, D[3][4] = 0
26     D[4][0] = 0, D[4][1] = 0, D[4][2] = 0, D[4][3] = 0, D[4][4] = 1
27     D[5][0] = 0, D[5][1] = 0, D[5][2] = 0, D[5][3] = 0, D[5][4] = 0
28     K[0][0] = 1, K[0][1] = 1, K[0][2] = 1, K[0][3] = 1, K[0][4] = 1
29     K[1][0] = 1, K[1][1] = 0, K[1][2] = 0, K[1][3] = 0, K[1][4] = 0
30     K[2][0] = 0, K[2][1] = 1, K[2][2] = 0, K[2][3] = 0, K[2][4] = 0
31     K[3][0] = 0, K[3][1] = 0, K[3][2] = 1, K[3][3] = 0, K[3][4] = 0
32     K[4][0] = 0, K[4][1] = 0, K[4][2] = 0, K[4][3] = 1, K[4][4] = 0
33     K[5][0] = 0, K[5][1] = 0, K[5][2] = 0, K[5][3] = 0, K[5][4] = 1
34     for(int i = 1; i <= n; i <= 1){
35         if(i & n) D = mul(K, D);
36         K = mul(K, K);
37     }
38     return D[0][0];
39 }
40 signed main(){
41     int n;
42     cin >> n;
43     cout << dice(n);
44     return 0;
45 }
```

## Graph Paths I (<https://cses.fi/problemset/task/1723>)

矩阵快速幂

```
1 #include <bits/stdc++.h>
2 #define int long long
3 #define matrix array<array<int, 101>, 101>
4 using namespace std;
5 const int mod = 1e9 + 7;
6 matrix G;
7 matrix mul(matrix A, matrix B, int n){
8     matrix C;
9     for(int i = 1; i <= n; i++){
10         for(int j = 1; j <= n; j++){
11             C[i][j] = 0;
12             for(int k = 1; k <= n; k++){
13                 C[i][j] += (A[i][k] * B[k][j]) % mod;
14             }
15             C[i][j] %= mod;
16         }
17     }
18     return C;
19 }
20 int walk(int n, int k){
21     matrix W;
22     for(int i = 1; i <= n; i++){
23         for(int j = 1; j <= n; j++){
24             W[i][j] = 0;
25         }
26         W[i][i] = 1;
27     }
28     for(int i = 1; i <= k; i <= n){
29         if(i & k) W = mul(G, W, n);
30         G = mul(G, G, n);
31     }
32     return W[n][1];
33 }
34 signed main(){
35     int n, m, k, a, b;
36     cin >> n >> m >> k;
37     while(m--){
38         cin >> a >> b;
39         G[b][a]++;
40     }
41     cout << walk(n, k);
42     return 0;
43 }
```

## Graph Paths II (<https://cses.fi/problemset/task/1724>)

矩阵

```
1 #include <bits/stdc++.h>
2 #define int long long
3 #define matrix array<array<int, 101>, 101>
4 using namespace std;
5 matrix G, W;
6 int min(int a, int b){
7     if(a < 0) return b;
8     if(b < 0) return a;
9     return a < b? a : b;
10 }
11 matrix mul(matrix A, matrix B, int n){
12     matrix C;
13     for(int i = 1; i <= n; i++){
14         for(int j = 1; j <= n; j++){
15             C[i][j] = -1;
16             for(int k = 1; k <= n; k++){
17                 if(A[i][k] < 0 || B[k][j] < 0) continue;
18                 C[i][j] = min(C[i][j], A[i][k] + B[k][j]);
19             }
20         }
21     }
22     return C;
23 }
24 int walk(int n, int k){
25     k--;
26     for(int i = 1; i <= k; i <= 1){
27         if(i & k) W = mul(G, W, n);
28         G = mul(G, G, n);
29     }
30     return W[n][1];
31 }
32 signed main(){
33     int n, m, k, a, b, c;
34     cin >> n >> m >> k;
35     for(int i = 1; i <= n; i++){
36         for(int j = 1; j <= n; j++){
37             G[i][j] = -1;
38             W[i][j] = -1;
39         }
40     }
41     while(m--){
42         cin >> a >> b >> c;
43         G[b][a] = min(G[b][a], c);
44         W[b][a] = min(W[b][a], c);
45     }
46     cout << walk(n, k);
47     return 0;
48 }
```

## Dice Probability (<https://cses.fi/problemset/task/1725>)

機率 DP

```
1 #include <bits/stdc++.h>
2 using namespace std;
3 array<array<double, 604>, 104> dp;
4 signed main(){
5     int n, a, b;
6     double all = 0, ans = 0;
7     cin >> n >> a >> b;
8     dp[0][0] = 1;
9     for(int i = 1; i <= n; i++){
10         for(int j = i; j <= 6 * i; j++){
11             for(int k = 1; k <= 6; k++){
12                 if(k > j) break;
13                 dp[i][j] += dp[i - 1][j - k];
14             }
15         }
16     }
17     for(int i = 1; i <= 6 * n; i++){
18         all += dp[n][i];
19         if(i >= a && i <= b) ans += dp[n][i];
20     }
21     cout << fixed << setprecision(6) << ans / all;
22     return 0;
23 }
```

## Moving Robots (<https://cses.fi/problemset/task/1726>)

機率 DP



```
1 #include <bits/stdc++.h>
2 using namespace std;
3 array<array<double, 8>, 8> cnt, dp, tmp, E;
4 void build(){
5     for(int i = 0; i < 8; i++){
6         for(int j = 0; j < 8; j++){
7             E[i][j] = 1;
8             if(i != 0) cnt[i][j] += 1;
9             if(i != 7) cnt[i][j] += 1;
10            if(j != 0) cnt[i][j] += 1;
11            if(j != 7) cnt[i][j] += 1;
12        }
13    }
14 }
15 void reset(){
16     for(int i = 0; i < 8; i++){
17         for(int j = 0; j < 8; j++){
18             dp[i][j] = 0;
19         }
20     }
21 }
22 void clear(){
23     for(int i = 0; i < 8; i++){
24         for(int j = 0; j < 8; j++){
25             tmp[i][j] = 0;
26         }
27     }
28 }
29 void move(int i, int j){
30     if(i != 0) tmp[i - 1][j] += dp[i][j] / cnt[i][j];
31     if(i != 7) tmp[i + 1][j] += dp[i][j] / cnt[i][j];
32     if(j != 0) tmp[i][j - 1] += dp[i][j] / cnt[i][j];
33     if(j != 7) tmp[i][j + 1] += dp[i][j] / cnt[i][j];
34 }
35 void trans(){
36     for(int i = 0; i < 8; i++){
37         for(int j = 0; j < 8; j++){
38             dp[i][j] = tmp[i][j];
39         }
40     }
41 }
42 signed main(){
43     int n;
44     double ans = 0;
45     cin >> n;
46     build();
47     for(int x = 0; x < 8; x++){
48         for(int y = 0; y < 8; y++){
49             reset();
50             dp[x][y] = 1;
51             for(int k = 0; k < n; k++){
52                 clear();
53                 for(int i = 0; i < 8; i++) {
```

```

54         for(int j = 0; j < 8; j++){
55             move(i, j);
56         }
57     }
58     trans();
59 }
60     for(int i = 0; i < 8; i++){
61         for(int j = 0; j < 8; j++){
62             E[i][j] *= (1.0 - dp[i][j]);
63         }
64     }
65 }
66 }
67     for(int i = 0; i < 8; i++){
68         for(int j = 0; j < 8; j++){
69             ans += E[i][j];
70         }
71     }
72     cout << fixed << setprecision(6) << ans;
73 }
74 }
```

## Candy Lottery (<https://cses.fi/problemset/task/1727>)

機率

```

1 #include <bits/stdc++.h>
2 using namespace std;
3
4 signed main(){
5     int n;
6     double k, ans = 0, u, d;
7     cin >> n >> k;
8     for(double i = 1; i <= k; i += 1){
9         u = d = 1;
10        for(int j = 0; j < n; j++){
11            u *= i / k;
12            d *= (i - 1) / k;
13        }
14        ans += i * (u - d);
15    }
16    cout << fixed << setprecision(6) << ans;
17 }
18 }
```

## Inversion Probability (<https://cses.fi/problemset/task/1728>)

機率

```

1 #include <bits/stdc++.h>
2 using namespace std;
3 array<double, 104> R;
4 signed main(){
5     int n;
6     double ans = 0;
7     cin >> n;
8     for(int i = 0; i < n; i++){
9         cin >> R[i];
10    }
11    for(int i = 0; i < n; i++){
12        for(int j = i + 1; j < n; j++){
13            for(double k = 2; k <= R[i]; k += 1){
14                ans += min(k - 1, R[j]) / R[j] / R[i];
15            }
16        }
17    }
18    cout << fixed << setprecision(6) << ans;
19    return 0;
20 }
```

## Stick Game (<https://cses.fi/problemset/task/1729>)

賽局 DP

```

1 #include <bits/stdc++.h>
2 using namespace std;
3 array<int, 104> P;
4 array<bool, 1000004> dp;
5 signed main(){
6     int n, k;
7     cin >> n >> k;
8     for(int i = 0; i < k; i++) cin >> P[i];
9     sort(P.begin(), P.begin() + k);
10    for(int i = 1; i <= n; i++){
11        for(int p : P){
12            if(p > i || !p) break;
13            dp[i] |= !dp[i - p];
14        }
15        cout << (dp[i] ? "W" : "L");
16    }
17    return 0;
18 }
```

## Nim Game I (<https://cses.fi/problemset/task/1730>)

賽局

```

1 #include <bits/stdc++.h>
2 using namespace std;
3
4 signed main(){
5     int t, n, x, ans;
6     cin >> t;
7     while(t--){
8         cin >> n;
9         ans = 0;
10        while(n--){
11            cin >> x;
12            ans ^= x;
13        }
14        cout << (ans? "first\n" : "second\n");
15    }
16    return 0;
17 }
```

## Nim Game II (<https://cses.fi/problemset/task/1098>)

賽局

```

1 #include <bits/stdc++.h>
2 using namespace std;
3
4 signed main(){
5     int t, n, x, ans;
6     cin >> t;
7     while(t--){
8         cin >> n;
9         ans = 0;
10        while(n--){
11            cin >> x;
12            ans ^= (x % 4);
13        }
14        cout << (ans? "first\n" : "second\n");
15    }
16    return 0;
17 }
```

## Stair Game (<https://cses.fi/problemset/task/1099>)

賽局

```

1 #include <bits/stdc++.h>
2 using namespace std;
3
4 signed main(){
5     int t, n, p, ans;
6     cin >> t;
7     while(t--){
8         cin >> n;
9         ans = 0;
10        for(int i = 1; i <= n; i++){
11            cin >> p;
12            if(~i & 1) ans ^= p;
13        }
14        cout << (ans? "first\n" : "second\n");
15    }
16    return 0;
17 }
```

## Grundy's Game (<https://cses.fi/problemset/task/2207>)

```

1 #include <bits/stdc++.h>
2 using namespace std;
3 array<int, 2004> SG;
4 int mex(set<int> &S){
5     for(int i = 0; i <= 2000; i++){
6         if(S.find(i) == S.end()) return i;
7     }
8 }
9 void build(int n){
10    set<int> S;
11    for(int i = 1; i <= n; i++){
12        S.clear();
13        for(int j = 1; j < i; j++){
14            if(j != i - j) S.insert(SG[i - j] ^ SG[j]);
15        }
16        SG[i] = mex(S);
17    }
18 }
19 signed main(){
20     int t, n;
21     cin >> t;
22     build(2000);
23     while(t--){
24         cin >> n;
25         if(n > 2000) cout << "first\n";
26         else cout << (SG[n]? "first\n" : "second\n");
27     }
28     return 0;
29 }
```

## Another Game (<https://cses.fi/problemset/task/2208>)

賽局

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

## String Algorithms

---

### Word Combinations (<https://cses.fi/problemset/task/1731>)

Trie DP

```
1 #include <bits/stdc++.h>
2 #define int long long
3 using namespace std;
4 const int mod = 1e9 + 7;
5 int p = 0;
6 string S;
7 array<array<int, 26>, 1000004> trie;
8 array<int, 1000004> cnt;
9 array<int, 5004> dp;
10 void update(string s){
11     int u = 0;
12     for(int i = 0; i < s.size(); i++){
13         if(!trie[u][s[i] - 'a']) trie[u][s[i] - 'a'] = ++p;
14         u = trie[u][s[i] - 'a'];
15     }
16     cnt[u]++;
17 }
18 int query(int i){
19     int u = 0, ans = 0;
20     for(; i < S.size(); i++){
21         if(!trie[u][S[i] - 'a']) return ans;
22         u = trie[u][S[i] - 'a'];
23         ans += (cnt[u] * dp[i + 1]) % mod;
24         ans %= mod;
25     }
26     return ans;
27 }
28 signed main(){
29     int k;
30     string K;
31     cin >> S >> k;
32     while(k--){
33         cin >> K;
34         update(K);
35     }
36     dp[S.size()] = 1;
37     for(int i = S.size() - 1; i >= 0; i--){
38         dp[i] += query(i);
39     }
40     cout << dp[0];
41     return 0;
42 }
```

## String Matching (<https://cses.fi/problemset/task/1753>)

KMP

```
1 #include <bits/stdc++.h>
2 using namespace std;
3 array<int, 1000004> F;
4 void build(string T){
5     int p;
6     F[0] = -1;
7     for(int i = 1; i < T.size(); i++){
8         p = F[i - 1];
9         while(~p && T[i] != T[p + 1]) p = F[p];
10        if(T[i] == T[p + 1]) p++;
11        F[i] = p;
12    }
13 }
14 int match(string T, string S){
15     int p = -1, cnt = 0;
16     for(int i = 0; i < S.size(); i++){
17         while(~p && S[i] != T[p + 1]) p = F[p];
18         if(S[i] == T[p + 1]) p++;
19         if(p + 1 == T.size()) cnt++, p = F[p];
20     }
21     return cnt;
22 }
23 signed main(){
24     string S, T;
25     cin >> S >> T;
26     build(T);
27     cout << match(T, S);
28     return 0;
29 }
```

## Finding Borders (<https://cses.fi/problemset/task/1732>)

Z

```
1 #include <bits/stdc++.h>
2 using namespace std;
3 array<int, 1000004> Z;
4 signed main(){
5     string S;
6     int l = 0, r = 0;
7     cin >> S;
8     Z[0] = S.size();
9     for(int i = 1; i < S.size(); i++){
10         if(i + Z[i - l] <= r) Z[i] = Z[i - l];
11         else{
12             l = i;
13             if(i > r) r = i;
14             while(r < S.size() && S[r] == S[r - l]) r++;
15             r--;
16             Z[i] = r - l + 1;
17         }
18     }
19     for(int i = S.size() - 1; i > 0; i--){
20         if(Z[i] == .size() - i) cout << Z[i] << " ";
21     }
22     return 0;
23 }
```

## Finding Periods (<https://cses.fi/problemset/task/1733>)

Z

```

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

## Minimal Rotation (<https://cses.fi/problemset/task/1110>)

Booth

```

1 #include <bits/stdc++.h>
2 using namespace std;
3 string LMS(string S){
4     int n = S.size(), i = 0, j = 1, k;
5     S += S;
6     while(i < n && j < n){
7         k = 0;
8         while(S[i + k] == S[j + k]) k++;
9         if(S[i + k] <= S[j + k]) j += k + 1;
10        else i += k + 1;
11        if(i == j) j++;
12    }
13    i = i < n? i : j;
14    return S.substr(i, n);
15 }
16 signed main(){
17     string S;
18     cin >> S;
19     cout << LMS(S);
20     return 0;
21 }
```

## Longest Palindrome (<https://cses.fi/problemset/task/1111>)

Manacher

```

1 #include <bits/stdc++.h>
2 #define pb push_back
3 #define mid (l + r) / 2
4 using namespace std;
5 array<int, 2000004> P;
6 string LPS(string T){
7     string S, A;
8     int l = 0, r = 0, lng = 1, ans = 0;
9     for(int i = 0; i <= 2 * T.size(); i++){
10         if(i & 1) S += T[i / 2];
11         else S += '#';
12     }
13     P[0] = 1;
14     for(int i = 1; i < S.size(); i++){
15         P[i] = max(min(r - i, P[2 * mid - i]), 1);
16         while(i >= P[i] && i + P[i] < S.size() && S[i - P[i]] == S[i + P[i]]){
17             l = i - P[i];
18             r = i + P[i];
19             P[i]++;
20         }
21         if(P[i] > lng || (P[i] == lng && i & 1)){
22             lng = P[i];
23             ans = i;
24         }
25     }
26     for(int i = ans - lng + 1; i < ans + lng; i++){
27         if(i & 1) A += S[i];
28     }
29     return A;
30 }
31 signed main(){
32     string S;
33     cin >> S;
34     cout << LPS(S);
35     return 0;
36 }
```

## Required Substring (<https://cses.fi/problemset/task/1112>)

KMP 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> F;
6 array<array<int, 104>, 1004> dp;
7 void KMP(string &S){
8     int p = 0;
9     for(int i = 2; i < S.size(); i++){
10         while(p && S[i - 1] != S[p]) p = F[p];
11         if(S[i - 1] == S[p]) p++;
12         F[i] = p;
13     }
14 }
15 int DP(string &S, int n, int m){
16     int p;
17     dp[0][0] = 1;
18     for(int i = 1; i <= m; i++){
19         for(int j = 0; j < n; j++){
20             for(char k = 'A'; k <= 'Z'; k++){
21                 p = j;
22                 while(p && S[p] != k) p = F[p];
23                 if(S[p] == k) p++;
24                 dp[i][p] += dp[i - 1][j];
25                 dp[i][p] %= mod;
26             }
27         }
28         dp[i][n] += dp[i - 1][n] * 26;
29         dp[i][n] %= mod;
30     }
31     return dp[m][n];
32 }
33 signed main(){
34     int n, m;
35     string S;
36     cin >> m >> S;
37     n = S.size();
38     KMP(S);
39     cout << DP(S, n, m);
40     return 0;
41 }
```

## Palindrome Queries (<https://cses.fi/problemset/task/2420>)

Hash 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 const int mod = 1e9 + 7;
8 array<int, 200004> H;
9 array<int, 800004> F, B;
10 void ha(int n){
11     H[0] = 1;
12     for(int i = 1; i <= n; i++){
13         H[i] = (H[i - 1] * 29) % mod;
14     }
15 }
16 void pull(int p, int l, int r){
17     F[p] = (F[lc] + H[mid - l + 1] * F[rc]) % mod;
18     B[p] = (B[rc] + H[r - mid] * B[lc]) % mod;
19 }
20 void update(int p, int c, int x, int l, int r){
21     if(c < l || c > r) return;
22     if(l == r){
23         F[p] = B[p] = x;
24         return;
25     }
26     update(lc, c, x, l, mid);
27     update(rc, c, x, mid + 1, r);
28     pull(p, l, r);
29 }
30 int query(int p, int ql, int qr, int l, int r, bool t){
31     if(ql > r || qr < l) return 0;
32     if(t){
33         if(ql <= l && qr >= r) return F[p];
34         return (query(lc, ql, qr, l, mid, t) + H[max(0ll, mid - max
35     }else{
36         if(ql <= l && qr >= r) return B[p];
37         return (query(rc, ql, qr, mid + 1, r, t) + H[max(0ll, min(r
38     }
39 }
40 signed main(){
41     int n, q, t, l, r, k, f, b;
42     char x;
43     cin >> n >> q;
44     ha(n);
45     for(int i = 1; i <= n; i++){
46         cin >> x;
47         update(1, i, x - 'a', 1, n);
48     }
49     while(q--){
50         cin >> t;
51         if(t == 1){
52             cin >> k >> x;
53             update(1, k, x - 'a', 1, n);
54         }
55     }
56 }
```

```
54     }else{
55         cin >> l >> r;
56         cout << (query(1, l, r, 1, n, 1) == query(1, l, r, 1, n,
57             })
58     }
59     return 0;
60 }
```

## Finding Patterns (<https://cses.fi/problemset/task/2102/>)

Suffix Array



```
1 #include <bits/stdc++.h>
2 #define pb push_back
3 #define mid (l + r) / 2
4 using namespace std;
5 array<int, 100004> SA, RNK, F, L;
6 array<vector<int>, 100004> buk;
7 void sort(array<int, 100004> &A, int n){
8     int cnt = 0;
9     for(int i = 0; i < n; i++){
10         buk[A[SA[i]]].pb(SA[i]);
11     }
12     for(int i = 0; i < max(n, 26); i++){
13         for(int x : buk[i]){
14             SA[cnt++] = x;
15         }
16         buk[i].clear();
17     }
18 }
19 void suf(string &S){
20     int n = S.size(), ff = -1, ll = -1, cnt = -1;
21     for(int i = 0; i < n; i++){
22         F[i] = S[i] - 'a';
23         SA[i] = i;
24     }
25     sort(F, n);
26     for(int i = 0; i < n; i++){
27         if(F[SA[i]] == ff && L[SA[i]] == ll) RNK[SA[i]] = cnt;
28         else RNK[SA[i]] = ++cnt;
29         ff = F[SA[i]], ll = L[SA[i]];
30     }
31     for(int j = 1; j < n; j <= 1){
32         cnt = ff = ll = -1;
33         for(int i = 0; i < n; i++){
34             F[i] = RNK[i];
35             L[i] = i + j < n? RNK[i + j] : 0;
36         }
37         sort(L, n);
38         sort(F, n);
39         for(int i = 0; i < n; i++){
40             if(F[SA[i]] == ff && L[SA[i]] == ll) RNK[SA[i]] = cnt;
41             else RNK[SA[i]] = ++cnt;
42             ff = F[SA[i]], ll = L[SA[i]];
43         }
44     }
45 }
46 bool cmp(string &T, string &S, int k){
47     for(int i = 0; i < T.size() && i + k < S.size(); i++){
48         if(T[i] < S[k + i]) return 1;
49         else if(T[i] > S[k + i]) return 0;
50     }
51     if(T.size() > S.size() - k) return 0;
52     return 1;
53 }
```

```
54     bool BS(string &S, string &T){  
55         int l = 0, r = S.size() - 1;  
56         while(l != r){  
57             if(cmp(T, S, SA[mid])) r = mid;  
58             else l = mid + 1;  
59         }  
60         if(T.size() > S.size() - SA[l]) return 0;  
61         for(int i = 0; i < T.size(); i++){  
62             if(T[i] != S[SA[l] + i]) return 0;  
63         }  
64         return 1;  
65     }  
66     signed main(){  
67         int k;  
68         string S, T;  
69         cin >> S >> k;  
70         suf(S);  
71         while(k--){  
72             cin >> T;  
73             cout << (BS(S, T)? "YES\n" : "NO\n");  
74         }  
75         return 0;  
76     }
```

## Counting Patterns (<https://cses.fi/problemset/task/2103/>)

Suffix Array



```
1 #include <bits/stdc++.h>
2 #define pb push_back
3 #define mid (l + r) / 2
4 using namespace std;
5 array<int, 100004> SA, RNK, F, L;
6 array<vector<int>, 100004> buk;
7 void sort(array<int, 100004> &A, int n){
8     int cnt = 0;
9     for(int i = 0; i < n; i++){
10         buk[A[SA[i]]].pb(SA[i]);
11     }
12     for(int i = 0; i < max(n, 26); i++){
13         for(int x : buk[i]){
14             SA[cnt++] = x;
15         }
16         buk[i].clear();
17     }
18 }
19 void suf(string &S){
20     int n = S.size(), cnt = -1, ff = -1, ll = -1;
21     for(int i = 0; i < n; i++){
22         SA[i] = i;
23         F[i] = S[i] - 'a';
24     }
25     sort(F, n);
26     for(int i = 0; i < n; i++){
27         if(F[SA[i]] == ff && L[SA[i]] == ll) RNK[SA[i]] = cnt;
28         else RNK[SA[i]] = ++cnt;
29         ff = F[SA[i]], ll = L[SA[i]];
30     }
31     for(int j = 1; j < n; j <= 1){
32         cnt = ff = ll = -1;
33         for(int i = 0; i < n; i++){
34             F[i] = RNK[i];
35             L[i] = i + j < n? RNK[i + j] : 0;
36         }
37         sort(L, n);
38         sort(F, n);
39         for(int i = 0; i < n; i++){
40             if(F[SA[i]] == ff && L[SA[i]] == ll) RNK[SA[i]] = cnt;
41             else RNK[SA[i]] = ++cnt;
42             ff = F[SA[i]], ll = L[SA[i]];
43         }
44     }
45 }
46 bool cmp(string &S, string &T, int k, bool t){
47     for(int i = 0; i < T.size() && i + k < S.size(); i++){
48         if(T[i] < S[i + k]) return 1;
49         else if(T[i] > S[i + k]) return 0;
50     }
51     if(T.size() > S.size() - k) return 0;
52     if(t) return 0;
53     else return 1;
```

```
54 }
55 int BS(string &S, string &T, bool t){
56     int l = 0, r = S.size() - 1;
57     while(l != r){
58         if(cmp(S, T, SA[mid], t)) r = mid;
59         else l = mid + 1;
60     }
61     return l;
62 }
63 signed main(){
64     int k;
65     string S, T;
66     cin >> S >> k;
67     S += '~';
68     suf(S);
69     while(k--){
70         cin >> T;
71         cout << BS(S, T, 1) - BS(S, T, 0) << "\n";
72     }
73     return 0;
74 }
```

## Pattern Positions (<https://cses.fi/problemset/task/2104>)



```
1 #include <bits/stdc++.h>
2 #define pb push_back
3 #define mid ((l + r) >> 1)
4 #define lc (p << 1)
5 #define rc ((p << 1) | 1)
6 using namespace std;
7 array<int, 400004> seg;
8 array<int, 100004> SA, RNK, F, L;
9 array<vector<int>, 100004> buk;
10 void update(int p, int c, int x, int l, int r){
11     if(c < l || c > r) return;
12     if(l == r){
13         seg[p] = x;
14         return;
15     }
16     update(lc, c, x, l, mid);
17     update(rc, c, x, mid + 1, r);
18     seg[p] = min(seg[lc], seg[rc]);
19 }
20 int query(int p, int ql, int qr, int l, int r){
21     if(ql > r || qr < l) return 1e9;
22     if(ql <= l && qr >= r) return seg[p];
23     return min(query(lc, ql, qr, l, mid), query(rc, ql, qr, mid + 1,
24 });
25 void sort(array<int, 100004> &A, int n){
26     int cnt = 0;
27     for(int i = 0; i < n; i++){
28         buk[A[SA[i]]].pb(SA[i]);
29     }
30     for(int i = 0; i < max(n, 26); i++){
31         for(int x : buk[i]){
32             SA[cnt++] = x;
33         }
34         buk[i].clear();
35     }
36 }
37 void suf(string &S){
38     int n = S.size(), cnt = -1, ff = -1, ll = -1;
39     for(int i = 0; i < n; i++){
40         SA[i] = i;
41         F[i] = S[i] - 'a';
42     }
43     sort(F, n);
44     for(int i = 0; i < n; i++){
45         if(F[SA[i]] == ff && L[SA[i]] == ll) RNK[SA[i]] = cnt;
46         else RNK[SA[i]] = ++cnt;
47         ff = F[SA[i]], ll = L[SA[i]];
48     }
49     for(int j = 1; j < n; j <= 1){
50         cnt = ll = ff = -1;
51         for(int i = 0; i < n; i++){
52             F[i] = RNK[i];
53             L[i] = i + j < n? RNK[i + j] : 0;
```

```

54     }
55     sort(L, n);
56     sort(F, n);
57     for(int i = 0; i < n; i++){
58         if(F[SA[i]] == ff && L[SA[i]] == ll) RNK[SA[i]] = cnt;
59         else RNK[SA[i]] = ++cnt;
60         ff = F[SA[i]], ll = L[SA[i]];
61     }
62 }
63 for(int i = 0; i < n; i++){
64     update(1, i, SA[i], 0, S.size() - 1);
65 }
66 }
67 bool cmp(string &S, string &T, int k, bool t){
68     for(int i = 0; i < T.size() && i + k < S.size(); i++){
69         if(T[i] < S[i + k]) return 1;
70         else if(T[i] > S[i + k]) return 0;
71     }
72     if(T.size() > S.size() - k) return 0;
73     return t ^ 1;
74 }
75 int BS(string &S, string &T, bool t){
76     int l = 0, r = S.size() - 1;
77     while(l != r){
78         if(cmp(S, T, SA[mid], t)) r = mid;
79         else l = mid + 1;
80     }
81     return l;
82 }
83 int pos(string &S, string &T){
84     if(BS(S, T, 1) == BS(S, T, 0)) return -1;
85     return query(1, BS(S, T, 0), BS(S, T, 1) - 1, 0, S.size() - 1)
86 }
87 signed main(){
88     int k;
89     string S, T;
90     cin >> S >> k;
91     S += '~';
92     for(int &s : seg) s = 1e9;
93     suf(S);
94     while(k--){
95         cin >> T;
96         cout << pos(S, T) << "\n";
97     }
98     return 0;
99 }
```

## Distinct Substrings (<https://cses.fi/problemset/task/2105>)

Suffix Array



```
1 #include <bits/stdc++.h>
2 #define pb push_back
3 #define int long long
4 using namespace std;
5 array<int, 100004> SA, RNK, F, L, LCP;
6 array<vector<int>, 100004> buk;
7 void sort(array<int, 100004> &A, int n){
8     int cnt = 0;
9     for(int i = 0; i < n; i++){
10         buk[A[SA[i]]].pb(SA[i]);
11     }
12     for(int i = 0; i < max(n, 26ll); i++){
13         for(int x : buk[i]){
14             SA[cnt++] = x;
15         }
16         buk[i].clear();
17     }
18 }
19 void suf(string &S){
20     int n = S.size(), cnt = -1, ff = -1, ll = -1;
21     for(int i = 0; i < n; i++){
22         SA[i] = n - i - 1;
23         F[i] = S[i] - 'a';
24     }
25     sort(F, n);
26     for(int i = 0; i < n; i++){
27         if(F[SA[i]] == ff && L[SA[i]] == ll) RNK[SA[i]] = cnt;
28         else RNK[SA[i]] = ++cnt;
29         ff = F[SA[i]], ll = L[SA[i]];
30     }
31     for(int j = 1; j < n; j <= 1){
32         cnt = ff = ll = -1;
33         for(int i = 0; i < n; i++){
34             F[i] = RNK[i];
35             L[i] = i + j < n? RNK[i + j] : 0;
36         }
37         sort(L, n);
38         sort(F, n);
39         for(int i = 0; i < n; i++){
40             if(F[SA[i]] == ff && L[SA[i]] == ll) RNK[SA[i]] = cnt;
41             else RNK[SA[i]] = ++cnt;
42             ff = F[SA[i]], ll = L[SA[i]];
43         }
44     }
45 }
46 int cp(string &S){
47     int n = S.size(), lcp = 0, k, sum = 0;
48     for(int i = 0; i < n; i++){
49         RNK[SA[i]] = i;
50     }
51     for(int i = 0; i < n; i++){
52         if(!RNK[i]) continue;
53         k = SA[RNK[i] - 1];
```

```
54     if(lcp) lcp--;
55     while(S[i + lcp] == S[k + lcp]) lcp++;
56     LCP[RNK[i]] = lcp;
57     sum += lcp;
58 }
59 return sum;
60 }
61 signed main(){
62     int n;
63     string S;
64     cin >> S;
65     n = S.size();
66     suf(S);
67     cout << n * (n + 1) / 2 - cp(S);
68     return 0;
69 }
```

## Repeating Substring (<https://cses.fi/problemset/task/2106>)

Suffix Array



```
1 #include <bits/stdc++.h>
2 #define pb push_back
3 using namespace std;
4 array<int, 100004> SA, RNK, F, L;
5 array<vector<int>, 100004> buk;
6 void sort(array<int, 100004> &A, int n){
7     int cnt = 0;
8     for(int i = 0; i < n; i++){
9         buk[A[SA[i]]].pb(SA[i]);
10    }
11    for(int i = 0; i < max(n, 26); i++){
12        for(int x : buk[i]){
13            SA[cnt++] = x;
14        }
15        buk[i].clear();
16    }
17 }
18 void suf(string &S){
19     int n = S.size(), cnt = -1, ff = -1, ll = -1;
20     for(int i = 0; i < n; i++){
21         SA[i] = n - i - 1;
22         F[i] = S[i] - 'a';
23     }
24     sort(F, n);
25     for(int i = 0; i < n; i++){
26         if(F[SA[i]] == ff && L[SA[i]] == ll) RNK[SA[i]] = cnt;
27         else RNK[SA[i]] = ++cnt;
28         ff = F[SA[i]], ll = L[SA[i]];
29     }
30     for(int j = 1; j < n; j <= 1){
31         cnt = ll = ff = -1;
32         for(int i = 0; i < n; i++){
33             F[i] = RNK[i];
34             L[i] = i + j < n? RNK[i + j] : 0;
35         }
36         sort(L, n);
37         sort(F, n);
38         for(int i = 0; i < n; i++){
39             if(F[SA[i]] == ff && L[SA[i]] == ll) RNK[SA[i]] = cnt;
40             else RNK[SA[i]] = ++cnt;
41             ff = F[SA[i]], ll = L[SA[i]];
42         }
43     }
44 }
45 string lcp(string &S){
46     int n = S.size(), cp = 0, k, lng = 0, ans;
47     for(int i = 0; i < n; i++){
48         RNK[SA[i]] = i;
49     }
50     for(int i = 0; i < n; i++){
51         if(!RNK[i]) continue;
52         k = SA[RNK[i] - 1];
53         if(cp) cp--;
```

```
54         while(S[i + cp] == S[k + cp]) cp++;
55         if(cp > lng){
56             lng = cp;
57             ans = i;
58         }
59     }
60     if(!lng) return "-1";
61     return S.substr(ans, lng);
62 }
63 signed main(){
64     string S;
65     cin >> S;
66     suf(S);
67     cout << lcp(S);
68     return 0;
69 }
```

## String Functions (<https://cses.fi/problemset/task/2107>)

Z KMP

```
1 #include <bits/stdc++.h>
2 using namespace std;
3 array<int, 1000004> Z, F;
4 void ZZZ(string &S){
5     int l = 0, r = 0;
6     Z[0] = 0;
7     for(int i = 1; i < S.size(); i++){
8         if(i + Z[i - l] < r) Z[i] = Z[i - l];
9         else{
10             l = i;
11             if(i > r) r = i;
12             while(r < S.size() && S[r] == S[r - l]) r++;
13             Z[i] = r - l;
14         }
15     }
16 }
17 void KMP(string &S){
18     int p;
19     F[0] = -1;
20     for(int i = 1; i < S.size(); i++){
21         p = F[i - 1];
22         while(~p && S[p + 1] != S[i]) p = F[p];
23         if(S[i] == S[p + 1]) p++;
24         F[i] = p;
25     }
26 }
27 signed main(){
28     string S;
29     cin >> S;
30     ZZZ(S);
31     for(int i = 0; i < S.size(); i++){
32         cout << Z[i] << " ";
33     }
34     cout << "\n";
35     KMP(S);
36     for(int i = 0; i < S.size(); i++){
37         cout << F[i] + 1 << " ";
38     }
39     return 0;
40 }
```

## Substring Order I (<https://cses.fi/problemset/task/2108>)

Suffix Array



```
1 #include <bits/stdc++.h>
2 #define int long long
3 #define pb push_back
4 #define mid (l + r) / 2
5 using namespace std;
6 array<int, 100004> SA, RNK, LCP, F, L;
7 array<vector<int>, 100004> buk;
8 void sort(array<int, 100004> &A, int n){
9     int cnt = 0;
10    for(int i = 0; i < n; i++){
11        buk[A[SA[i]]].pb(SA[i]);
12    }
13    for(int i = 0; i < max(n, 26ll); i++){
14        for(int x : buk[i]){
15            SA[cnt++] = x;
16        }
17        buk[i].clear();
18    }
19 }
20 void suf(string &S){
21     int n = S.size(), cnt = -1, ff = -1, ll = -1;
22     for(int i = 0; i < n; i++){
23         SA[i] = n - i - 1;
24         F[i] = S[i] - 'a';
25     }
26     sort(F, n);
27     for(int i = 0; i < n; i++){
28         if(F[SA[i]] == ff && L[SA[i]] == ll) RNK[SA[i]] = cnt;
29         else RNK[SA[i]] = ++cnt;
30         ff = F[SA[i]], ll = L[SA[i]];
31     }
32     for(int j = 1; j < n; j <= 1){
33         cnt = ff = ll = -1;
34         for(int i = 0; i < n; i++){
35             F[i] = RNK[i];
36             L[i] = i + j < n? RNK[i + j] : 0;
37         }
38         sort(L, n);
39         sort(F, n);
40         for(int i = 0; i < n; i++){
41             if(F[SA[i]] == ff && L[SA[i]] == ll) RNK[SA[i]] = cnt;
42             else RNK[SA[i]] = ++cnt;
43             ff = F[SA[i]], ll = L[SA[i]];
44         }
45     }
46 }
47 void lcp(string &S){
48     int n = S.size(), cp = 0, k;
49     for(int i = 0; i < n; i++){
50         RNK[SA[i]] = i;
51     }
52     for(int i = 0; i < n; i++){
53         if(!RNK[i]) continue;
```

```
54         k = SA[RNK[i] - 1];
55         if(cp) cp--;
56         while(S[i + cp] == S[k + cp]) cp++;
57         LCP[RNK[i]] = cp;
58     }
59 }
60 string see(string &S, int k){
61     int n = S.size(), i;
62     for(i = 0; i < n; i++){
63         if(k - (n - SA[i] - LCP[i]) <= 0) break;
64         k -= n - SA[i] - LCP[i];
65     }
66     return S.substr(SA[i], LCP[i] + k);
67 }
68 signed main(){
69     int k;
70     string S;
71     cin >> S >> k;
72     suf(S);
73     lcp(S);
74     cout << see(S, k);
75     return 0;
76 }
```

## Substring Order II (<https://cses.fi/problemset/task/2109>)



```
1 #include <bits/stdc++.h>
2 #define int long long
3 #define pb push_back
4 #define mid ((l + r) >> 1)
5 #define lc (p << 1)
6 #define rc ((p << 1) + 1)
7 using namespace std;
8 array<int, 100004> SA, RNK, F, L;
9 array<int, 400004> seg, tag;
10 array<vector<int>, 100004> buk;
11 void pull(int p){
12     seg[p] = seg[lc] + seg[rc];
13 }
14 void push(int p, int l, int r){
15     seg[lc] += (mid - l + 1) * tag[p];
16     seg[rc] += (r - mid) * tag[p];
17     tag[lc] += tag[p];
18     tag[rc] += tag[p];
19     tag[p] = 0;
20 }
21 void update(int p, int ql, int qr, int v, int l, int r){
22     if(ql > r || qr < l) return;
23     if(ql <= l && qr >= r){
24         seg[p] += (r - l + 1) * v;
25         tag[p] += v;
26         return;
27     }
28     push(p, l, r);
29     update(lc, ql, qr, v, l, mid);
30     update(rc, ql, qr, v, mid + 1, r);
31     pull(p);
32 }
33 int query(int p, int ql, int qr, int l, int r){
34     if(ql > r || qr < l) return 0;
35     if(ql <= l && qr >= r) return seg[p];
36     push(p, l, r);
37     return query(lc, ql, qr, l, mid) + query(rc, ql, qr, mid + 1,
38 }
39 void sort(array<int, 100004> &A, int n){
40     int cnt = 0;
41     for(int i = 0; i < n; i++){
42         buk[A[SA[i]]].pb(SA[i]);
43     }
44     for(int i = 0; i < max(n, 26ll); i++){
45         for(int x : buk[i]) SA[cnt++] = x;
46         buk[i].clear();
47     }
48 }
49 void suf(string &S){
50     int n = S.size(), cnt, ff, ll;
51     for(int i = 0; i < n; i++){
52         SA[i] = n - i - 1;
53         RNK[i] = F[i] = S[i] - 'a';
```

```
54     }
55     sort(F, n);
56     for(int k = 1; k < n; k <= 1){
57         cnt = ff = ll = -1;
58         for(int i = 0; i < n; i++){
59             F[i] = RNK[i];
60             L[i] = i + k < n? RNK[i + k] : 0;
61         }
62         sort(L, n);
63         sort(F, n);
64         for(int i = 0; i < n; i++){
65             if(F[SA[i]] == ff && L[SA[i]] == ll) RNK[SA[i]] = cnt;
66             else RNK[SA[i]] = ++cnt;
67             ff = F[SA[i]], ll = L[SA[i]];
68         }
69     }
70     for(int i = 0; i < n; i++){
71         update(1, i, i, n - SA[i], 0, n);
72     }
73 }
74 int bis(string &S, int t, int l, int r, char c){
75     int p = l - 1;
76     for(int i = 1 << 16; i > 0; i >>= 1){
77         if(p + i <= r && S[SA[p + i] + t] <= c) p += i;
78     }
79     return p;
80 }
81 char cha(string &S, int k, int t, int l, int r){
82     int p, n = S.size();
83     char cl = 'a', cr = 'z', cm;
84     while(cl != cr){
85         cm = (cl + cr) >> 1;
86         p = bis(S, t, l, r, cm);
87         if(query(1, l, p, 0, n) < k) cl = cm + 1;
88         else cr = cm;
89     }
90     return cl;
91 }
92 string see(string &S, int k){
93     int n = S.size(), l = 0, r = n - 1, t = 0, tmp;
94     char p;
95     string s;
96     while(k > 0){
97         p = cha(S, k, t, l, r);
98         tmp = l;
99         l = bis(S, t, l, r, p - 1) + 1;
100        k -= query(1, tmp, l - 1, 0, n);
101        r = bis(S, t, l, r, p);
102        update(1, l, r, -1, 0, n);
103        k -= r - l + 1;
104        s += p;
105        t++;
106    }
107 }
```

```
107     return s;
108 }
109 signed main(){
110     int k;
111     string S;
112     cin >> S >> k;
113     suf(S);
114     cout << see(S, k) << "\n";
115     return 0;
116 }
```

## Substring Distribution (<https://cses.fi/problemset/task/2110>)

Suffix Array



```
1 #include <bits/stdc++.h>
2 #define int long long
3 #define pb push_back
4 using namespace std;
5 array<int, 100004> SA, RNK, F, L, sum;
6 array<vector<int>, 100004> buk;
7 void sort(array<int, 100004> &A, int n){
8     int cnt = 0;
9     for(int i = 0; i < n; i++){
10         buk[A[SA[i]]].pb(SA[i]);
11     }
12     for(int i = 0; i < max(n, 26ll); i++){
13         for(int x : buk[i]){
14             SA[cnt++] = x;
15         }
16         buk[i].clear();
17     }
18 }
19 void suf(string &S){
20     int n = S.size(), cnt, ff, ll;
21     for(int i = 0; i < n; i++){
22         SA[i] = n - i - 1;
23         RNK[i] = F[i] = S[i] - 'a';
24     }
25     sort(F, n);
26     for(int j = 1; j < n; j <= 1){
27         cnt = ff = ll = -1;
28         for(int i = 0; i < n; i++){
29             F[i] = RNK[i];
30             L[i] = i + j < n? RNK[i + j] : 0;
31         }
32         sort(L, n);
33         sort(F, n);
34         for(int i = 0; i < n; i++){
35             if(F[SA[i]] == ff && L[SA[i]] == ll) RNK[SA[i]] = cnt;
36             else RNK[SA[i]] = ++cnt;
37             ff = F[SA[i]], ll = L[SA[i]];
38         }
39     }
40 }
41 void lcp(string &S){
42     int n = S.size(), cp = 0, k;
43     for(int i = 0; i < n; i++) RNK[SA[i]] = i;
44     for(int i = 0; i < n; i++){
45         if(!RNK[i]){
46             sum[0]++;
47             sum[n - i]--;
48             continue;
49         }
50         k = SA[RNK[i] - 1];
51         if(cp) cp--;
52         while(S[i + cp] == S[k + cp]) cp++;
53         sum[cp]++;
54 }
```

```
54         sum[n - i]--;
55     }
56 }
57 signed main(){
58     int ans = 0;
59     string S;
60     cin >> S;
61     suf(S);
62     lcp(S);
63     for(int i = 0; i < S.size(); i++){
64         ans += sum[i];
65         cout << ans << " ";
66     }
67     return 0;
68 }
```

## Geometry

---

### Point Location Test (<https://cses.fi/problemset/task/2189>)

向量

```
1 #include <bits/stdc++.h>
2 #define int long long
3 using namespace std;
4 struct vec{
5     int x, y;
6     vec(int x, int y): x(x), y(y){}
7     vec operator+(vec v){
8         return vec(x + v.x, y + v.y);
9     }
10    vec operator-(vec v){
11        return vec(x - v.x, y - v.y);
12    }
13    int operator*(vec v){
14        return x * v.x + y * v.y;
15    }
16    int operator^(vec v){
17        return x * v.y - y * v.x;
18    }
19 };
20 signed main(){
21     int t, x1, y1, x2, y2, x3, y3, ans;
22     cin >> t;
23     while(t--){
24         cin >> x1 >> y1 >> x2 >> y2 >> x3 >> y3;
25         ans = vec(x2 - x1, y2 - y1) ^ vec(x3 - x1, y3 - y1);
26         if(ans > 0) cout << "LEFT\n";
27         else if(ans < 0) cout << "RIGHT\n";
28         else cout << "TOUCH\n";
29     }
30     return 0;
31 }
```

## Line Segment Intersection (<https://cses.fi/problemset/task/2190>)

向量

```
1 #include <bits/stdc++.h>
2 #define int long long
3 using namespace std;
4 struct vec{
5     int x, y;
6     vec(): x(0), y(0){}
7     vec(int x, int y): x(x), y(y){}
8     vec operator+(vec v){
9         return vec(x + v.x, y + v.y);
10    }
11    vec operator-(vec v){
12        return vec(x - v.x, y - v.y);
13    }
14    int operator*(vec v){
15        return x * v.x + y * v.y;
16    }
17    int operator^(vec v){
18        return x * v.y - y * v.x;
19    }
20 };
21 int cal(vec a, vec b, vec c, bool t){
22     int ans;
23     if(t) ans = (c - a) ^ (b - a);
24     else ans = (c - a) * (b - a);
25     if(ans == 0) return 0;
26     else if(ans > 0) return 1;
27     else return -1;
28 }
29 bool line(vec a, vec b, vec c, vec d){
30     bool ans = 1;
31     ans &= !cal(a, b, c, 1) && !cal(a, b, d, 1);
32     ans &= (a - c) * (b - d) > 0 && (a - d) * (b - c) > 0;
33     return ans;
34 }
35 bool inter(vec a, vec b, vec c, vec d){
36     bool ans = 1;
37     ans &= cal(a, b, c, 1) * cal(a, b, d, 1) <= 0;
38     ans &= cal(c, d, a, 1) * cal(c, d, b, 1) <= 0;
39     ans &= !line(a, b, c, d);
40     return ans;
41 }
42 signed main(){
43     int t, x1, y1, x2, y2, x3, y3, x4, y4;
44     cin >> t;
45     while(t--){
46         cin >> x1 >> y1 >> x2 >> y2 >> x3 >> y3 >> x4 >> y4;
47         cout << (inter(vec(x1, y1), vec(x2, y2), vec(x3, y3), vec(x
48         })
49         return 0;
50 }
```

## Polygon Area (<https://cses.fi/problemset/task/2191>)

向量 行列式

```
1 #include <bits/stdc++.h>
2 #define int long long
3 using namespace std;
4 struct vec{
5     int x, y;
6     vec(){} 
7     vec(int x, int y): x(x), y(y){}
8     vec operator+(vec v){
9         return vec(x + v.x, y + v.y);
10    }
11    vec operator-(vec v){
12        return vec(x - v.x, y - v.y);
13    }
14    int operator*(vec v){
15        return x * v.x + y * v.y;
16    }
17    int operator^(vec v){
18        return x * v.y - y * v.x;
19    }
20 };
21 array<vec, 1004> E;
22 int area(int n){
23     int ans = 0;
24     for(int i = 0; i < n; i++){
25         ans += E[i] ^ E[i + 1];
26     }
27     return abs(ans);
28 }
29 signed main(){
30     int n, x, y;
31     cin >> n;
32     for(int i = 0; i < n; i++){
33         cin >> x >> y;
34         E[i] = vec(x, y);
35     }
36     E[n] = E[0];
37     cout << area(n);
38     return 0;
39 }
```

## Point in Polygon (<https://cses.fi/problemset/task/2192>)

向量



```
1 #include <bits/stdc++.h>
2 #define int long long
3 using namespace std;
4 struct vec{
5     int x, y;
6     vec():{} 
7     vec(int x, int y): x(x), y(y){}
8     vec operator+(vec v){
9         return vec(x + v.x, y + v.y);
10    }
11    vec operator-(vec v){
12        return vec(x - v.x, y - v.y);
13    }
14    int operator*(vec v){
15        return x * v.x + y * v.y;
16    }
17    int operator^(vec v){
18        return x * v.y - y * v.x;
19    }
20 };
21 const vec inf = vec(1, 4e9);
22 array<vec, 1004> V;
23 int cal(vec a, vec b, vec c){
24     int ans;
25     ans = (c - a) ^ (b - a);
26     if(ans > 0) return 1;
27     else if(ans < 0) return -1;
28 }
29 int inter(vec a, vec b, vec c, vec d){
30     int abc, abd, cda, cdb;
31     abc = cal(a, b, c);
32     abd = cal(a, b, d);
33     cda = cal(c, d, a);
34     cdb = cal(c, d, b);
35     if(abc * abd < 0 && cda * cdb < 0) return 1;
36     else return 0;
37 }
38 bool bound(vec a, vec b, vec c){
39     if((a - b) ^ (c - b)) return 0;
40     if((a - b) * (c - b) > (c - b) * (c - b)) return 0;
41     if((a - b) * (c - b) < 0) return 0;
42     return 1;
43 }
44 int in(vec p, int n){
45     int cnt = 0;
46     bool on = 0;
47     vec end = p + inf;
48     for(int i = 0; i < n; i++){
49         cnt += inter(p, end, V[i], V[i + 1]);
50         on |= bound(p, V[i], V[i + 1]);
51     }
52     if(on) return 0;
53     if(cnt & 1) return 1;
```

```
54     else return -1;
55 }
56 signed main(){
57     int n, m, x, y, ans;
58     cin >> n >> m;
59     for(int i = 0; i < n; i++){
60         cin >> x >> y;
61         V[i] = vec(x, y);
62     }
63     V[n] = V[0];
64     while(m--){
65         cin >> x >> y;
66         ans = in(vec(x, y), n);
67         if(ans > 0) cout << "INSIDE\n";
68         else if(ans < 0) cout << "OUTSIDE\n";
69         else cout << "BOUNDARY\n";
70     }
71     return 0;
72 }
```

## Polygon Lattice Points (<https://cses.fi/problemset/task/2193>)

向量 皮克定理

```
1 #include <bits/stdc++.h>
2 #define int long long
3 using namespace std;
4 struct vec{
5     int x, y;
6     vec(): x(0), y(0){}
7     vec(int x, int y): x(x), y(y){}
8     vec operator+(vec v){
9         return vec(x + v.x, y + v.y);
10    }
11    vec operator-(vec v){
12        return vec(x - v.x, y - v.y);
13    }
14    int operator*(vec v){
15        return x * v.x + y * v.y;
16    }
17    int operator^(vec v){
18        return x * v.y - y * v.x;
19    }
20 };
21 array<vec, 100004> V;
22 int gcd(int a, int b){
23     return b? gcd(b, a % b) : a;
24 }
25 void point(int n){
26     int area = 0, on = 0, in;
27     vec tmp;
28     for(int i = 0; i < n; i++){
29         tmp = V[i] - V[i + 1];
30         area += V[i] ^ V[i + 1];
31         on += gcd(abs(tmp.x), abs(tmp.y));
32     }
33     area = abs(area);
34     in = (area - on + 2) / 2;
35     cout << in << " " << on;
36 }
37 signed main(){
38     int n, x, y;
39     cin >> n;
40     for(int i = 0; i < n; i++){
41         cin >> x >> y;
42         V[i] = vec(x, y);
43     }
44     V[n] = V[0];
45     point(n);
46     return 0;
47 }
```

## Minimum Euclidean Distance (<https://cses.fi/problemset/task/2194>)

掃描線

```

1 #include <bits/stdc++.h>
2 #define int long long
3 #define vec pair<int, int>
4 #define x first
5 #define y second
6 using namespace std;
7 array<vec, 200004> V;
8 int sq(int x){
9     return ceil(sqrt(x));
10 }
11 int dis(vec a, vec b){
12     return (a.x - b.x) * (a.x - b.x) + (a.y - b.y) * (a.y - b.y);
13 }
14 int mindis(int n){
15     int p = 0, d = 8e18;
16     set<vec> S;
17     for(int i = 0; i < n; i++){
18         while(V[p].x <= V[i].x - d){
19             S.erase({V[p].y, V[p].x});
20             p++;
21         }
22         for(auto it = S.upper_bound({V[i].y - sq(d), V[i].x}); it->
23             if(it == S.end()) break;
24             d = min(d, dis(*it, {V[i].y, V[i].x}));
25         }
26         S.insert({V[i].y, V[i].x});
27     }
28     return d;
29 }
30 signed main(){
31     int n, x, y;
32     cin >> n;
33     for(int i = 0; i < n; i++){
34         cin >> x >> y;
35         V[i] = {x, y};
36     }
37     sort(V.begin(), V.begin() + n);
38     cout << mindis(n);
39 }
40 }
```

## Convex Hull (<https://cses.fi/problemset/task/2195>)

向量 Monoton Stack



```
1 #include <bits/stdc++.h>
2 #define int long long
3 #define pb push_back
4 #define ppb pop_back
5 using namespace std;
6 struct vec{
7     int x, y;
8     vec(){}
9     vec(int x, int y): x(x), y(y){}
10    vec operator+(vec v){
11        return vec(x + v.x, y + v.y);
12    }
13    vec operator-(vec v){
14        return vec(x - v.x, y - v.y);
15    }
16    int operator*(vec v){
17        return x * v.x + y * v.y;
18    }
19    int operator^(vec v){
20        return x * v.y - y * v.x;
21    }
22 };
23 array<vec, 200004> V;
24 vector<vec> S;
25 bool cmp(vec a, vec b){
26     return a.x == b.x? a.y < b.y : a.x < b.x;
27 }
28 void print(vec v){
29     cout << v.x << " " << v.y << "\n";
30 }
31 bool comp(vec a, vec b){
32     if(a ^ b) return (a ^ b) > 0;
33     return (a * b) > 0;
34 }
35 void hull(int n, int s){
36     vec a, b;
37     for(int i = 0; i < n; i++){
38         while(S.size() > s){
39             b = S.back();
40             S.ppb();
41             a = S.back();
42             if(comp(V[i] - a, b - a)){
43                 S.pb(b);
44                 break;
45             }
46         }
47         S.pb(V[i]);
48     }
49     S.ppb();
50 }
51 signed main(){
52     int n, x, y;
53     cin >> n;
```

```
54     for(int i = 0; i < n; i++){
55         cin >> x >> y;
56         V[i] = vec(x, y);
57     }
58     sort(V.begin(), V.begin() + n, cmp);
59     hull(n, 1);
60     reverse(V.begin(), V.begin() + n);
61     hull(n, S.size() + 1);
62     cout << S.size() << "\n";
63     for(vec s : S) print(s);
64
65 }
```