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)

目錄

Range Queries

Static Range Sum Queries (https://cses.fi/problemset/task/1646)

前綴

```cpp
#include <bits/stdc++.h>
#define int long long
using namespace std;
array<int, 200004> S;
signed main(){
    int n, q, l, r;
    cin >> n >> q;
    for(int i = 1; i <= n; i++){
        cin >> S[i];
        S[i] += S[i - 1];
    }
    while(q--){
        cin >> l >> r;
        cout << S[r] - S[l - 1] << "\n";
    }
    return 0;
}
```

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

線段樹

```cpp
#include <bits/stdc++.h>
#define mid ((l + r) >> 1)
#define lc (p << 1)
#define rc ((p << 1) | 1)
using namespace std;
array<int, 800004> S;
void pull(int p){
S[p] = min(S[lc], S[rc]);
}
void update(int p, int c, int x, int l, int r){
    if(c > r || c < l) return;
    if(l == r){
        S[p] = x;
        return;
    }
    update(lc, c, x, l, mid);
    update(rc, c, x, mid + 1, r);
    pull(p);
}
int query(int p, int ql, int qr, int l, int r){
    if(ql > r || qr < l) return 1e9;
    if(ql <= l && qr >= r) return S[p];
    return min(query(lc, ql, qr, l, mid), query(rc, ql, qr, mid + 1
}
signed main(){
    int n, q, l, r, x;
    cin >> n >> q;
    for(int i = 1; i <= n; i++){
        cin >> x;
        update(1, i, x, 1, n);
    }
    while(q--){
        cin >> l >> r;
        cout << query(1, l, r, 1, n) << "\n";
    }
    return 0;
}
```


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

BIT

```cpp
#include <bits/stdc++.h>
#define int long long
using namespace std;
array<int, 200004> BIT, S;
void update(int p, int x){
    for(; p < 200004; p += p & -p){
        BIT[p] += x;
    }
}
int query(int p){
    int sum = 0;
    for(; p > 0; p -= p & -p){
        sum += BIT[p];
    }
    return sum;
}
signed main(){
    int n, q, t, l, r;
    cin >> n >> q;
    for(int i = 1; i <= n; i++){
        cin >> S[i];
        update(i, S[i]);
    }
    while(q--){
        cin >> t >> l >> r;
        if(t == 1){
            update(l, r - S[l]);
            S[l] = r;
        }else{
            cout << query(r) - query(l - 1) << "\n";
;        }
    }
    return 0;
}
```

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

線段樹

```cpp
#include <bits/stdc++.h>
#define mid ((l + r) >> 1)
#define lc (p << 1)
#define rc ((p << 1) | 1)
using namespace std;
array<int, 800004> S;
void pull(int p){
    S[p] = min(S[lc], S[rc]);
}
void update(int p, int c, int x, int l, int r){
    if(c < l || c > r) return;
    if(l == r){
        S[p] = x;
        return;
    }
    update(lc, c, x, l, mid);
    update(rc, c, x, mid + 1, r);
    pull(p);
}
int query(int p, int ql, int qr, int l, int r){
    if(l > qr || r < ql) return 1e9;
    if(ql <= l && qr >= r) return S[p];
    return min(query(lc, ql, qr, l, mid), query(rc, ql, qr, mid + 1
}
signed main(){
    int n, q, t, l, r;
    cin >> n >> q;
    for(int i = 1; i <= n; i++){
        cin >> t;
        update(1, i, t, 1, n);
    }
    while(q--){
        cin >> t >> l >> r;
        if(t == 1){
            update(1, l, r, 1, n);
        }else{
            cout << query(1, l, r, 1, n) << "\n";
        }
    }
    return 0;
}
```

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

前綴

```cpp
#include <bits/stdc++.h>
using namespace std;
array<int, 200004> S;
signed main(){
    int n, q, l, r;
    cin >> n >> q;
    for(int i = 1; i <= n; i++){
        cin >> S[i];
        S[i] ^= S[i - 1];
    }
    while(q--){
        cin >>l >> r;
        cout << (S[r] ^ S[l - 1]) << "\n";
    }
    return 0;
}
```

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

BIT

```cpp
#include <bits/stdc++.h>
#define int long long
using namespace std;
array<int, 200004> BIT;
void update(int p, int x){
    for(; p < 200004; p += p & -p){
        BIT[p] += x;
    }
}
int query(int p){
    int sum = 0;
    for(; p > 0; p -= p & -p){
        sum += BIT[p];
    }
    return sum;
}
signed main(){
    int n, q, t, l, r, x;
    cin >> n >> q;
    for(int i = 1; i <= n; i++){
        cin >> x;
        update(i, x - query(i - 1));
    }
    while(q--){
        cin >> t;
        if(t == 1){
            cin >> l >> r >> x;
            update(l, x);
            update(r + 1, -x);
        }else{
            cin >> x;
            cout << query(x) << "\n";
        }
    }
    return 0;
}
```

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

前綴

```cpp
#include <bits/stdc++.h>
using namespace std;
array<array<int, 1004>, 1004> F;
signed main(){
    int n, q, x1, x2, y1, y2;
    char f;
    cin >> n >> q;
    for(int i = 1; i <= n; i++){
        for(int j = 1; j <= n; j++){
            cin >> f;
            if(f == '*') F[i][j]++;
            F[i][j] += F[i - 1][j] + F[i][j - 1] - F[i - 1][j - 1];
        }
    }
    while(q--){
        cin >> x1 >> y1 >> x2 >> y2;
        cout << F[x2][y2] - F[x2][y1 - 1] - F[x1 - 1][y2] + F[x1 -
    }
    return 0;
}
```

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

線段樹　二分搜

```cpp
#include <bits/stdc++.h>
#define mid ((l + r) >> 1)
#define lc (p << 1)
#define rc ((p << 1) | 1)
using namespace std;
array<int, 800004> S;
void pull(int p){
    S[p] = max(S[lc], S[rc]);
}
void update(int p, int c, int x, int l, int r){
    if(c < l || c > r) return;
    if(l == r){
        S[p] += x;
        return;
    }
    update(lc, c, x, l, mid);
    update(rc, c, x, mid + 1, r);
    pull(p);
}
int query(int p, int x, int l, int r){
    if(S[p] < x) return 0;
    if(l == r) return l;
    if(S[lc] >= x) return query(lc, x, l, mid);
    else return query(rc, x, mid + 1, r);
}
signed main(){
    int n, m, h, r, p;
    cin >> n >> m;
    for(int i = 1; i <= n; i++){
        cin >> h;
        update(1, i, h, 1, n);
    }
    while(m--){
        cin >> r;
        p = query(1, r, 1, n);
        cout << p << " ";
        if(p) update(1, p, -r, 1, n);
    }
    return 0;
}
```

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

BIT 二分搜

```cpp
#include <bits/stdc++.h>
using namespace std;
array<int, 200004> BIT, S;
void update(int p, int x){
    for(; p < 200004; p += p & -p){
        BIT[p] += x;
    }
}
int find(int x){
    int sum = 0, p = 0;
    for(int i = 17; i >= 0; i--){
        if(p + (1 << i) < 200004 && sum + BIT[p + (1 << i)] < x){
            p += (1 << i);
            sum += BIT[p];
        }
    }
    return p + 1;
}
signed main(){
    int n, p;
    cin >> n;
    for(int i = 1; i <= n; i++){
        cin >> S[i];
        update(i, 1);
    }
    while(n--){
        cin >> p;
        cout << S[find(p)] << " ";
        update(find(p), -1);
    }
    return 0;
}
```

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

動態開點線段樹

```cpp
#include <bits/stdc++.h>
#define mid ((l + r) >> 1)
#define pb push_back
using namespace std;
struct node{
    int val, lc, rc;
};
array<int, 200004> P;
vector<node> S;
void pull(int p){
    S[p].val = S[S[p].lc].val + S[S[p].rc].val;
}
void update(int p, int c, int x, int l, int r){
    if(c < l || c > r) return;
    if(l == r){
        S[p].val += x;
        return;
    }
    if(!S[p].lc){
        S[p].lc = S.size();
        S.pb({0, 0, 0});
    }
    if(!S[p].rc){
        S[p].rc = S.size();
        S.pb({0, 0, 0});
    }
    update(S[p].lc, c, x, l, mid);
    update(S[p].rc, c, x, mid + 1, r);
    pull(p);
}
int query(int p, int ql, int qr, int l, int r){
    if(ql > r || qr < l || !p) return 0;
    if(ql <= l && qr >= r) return S[p].val;
    return query(S[p].lc, ql, qr, l, mid) + query(S[p].rc, ql, qr,
}
signed main(){
    cin.tie(0), cout.tie(0), ios::sync_with_stdio(0);
    int n, q, l, r;
    char t;
    S.pb({0, 0, 0});
    S.pb({0, 0, 0});
    cin >> n >> q;
    for(int i = 1; i <= n; i++){
        cin >> P[i];
        update(1, P[i], 1, 1, 1e9);
    }
    while(q--){
        cin >> t >> l >> r;
        if(t == '!'){
            update(1, P[l], -1, 1, 1e9);
            P[l] = r;
            update(1, P[l], 1, 1, 1e9);
        }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

```cpp
#include <bits/stdc++.h>
#define int long long
#define mid ((l + r) >> 1)
#define lc (p << 1)
#define rc ((p << 1) | 1)
#define pii pair<int, int>
#define ff first
#define ss second
using namespace std;
array<int, 800004> S, P;
void pull(int p){
    S[p] = S[lc] + S[rc];
    P[p] = max(S[lc] + P[rc], P[lc]);
}
void update(int p, int c, int x, int l, int r){
    if(c < l || c > r) return;
    if(l == r){
        S[p] = P[p] = x;
        return;
    }
    update(lc, c, x, l, mid);
    update(rc, c, x, mid + 1, r);
    pull(p);
}
pii query(int p, int ql, int qr, int l, int r){
    if(ql > r || qr < l) return {0, 0};
    if(ql <= l && qr >= r){
        return {P[p], S[p]};
    }
    pii ll, rr;
    ll = query(lc, ql, qr, l, mid);
    rr = query(rc, ql, qr, mid + 1, r);
    return {max(ll.ff, ll.ss + rr.ff), ll.ss + rr.ss};
}
signed main(){
    int n, q, t, l, r, x;
    cin >> n >> q;
    for(int i = 1; i <= n; i++){
        cin >> x;
        update(1, i, x, 1, n);
    }
    while(q--){
        cin >> t >> l >> r;
        if(t == 1){
            update(1, l, r, 1, n);
        }else{
            cout << max(0ll, query(1, l, r, 1, n).ff) << "\n";
        }
    }
    return 0;
}
```

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

線段樹

```cpp
#include <bits/stdc++.h>
#define mid ((l + r) >> 1)
#define lc (p << 1)
#define rc ((p << 1) | 1)
using namespace std;
array<int, 200004> P;
array<int, 800004> L, R;
void pull(int p){
    L[p] = min(L[lc], L[rc]);
    R[p] = min(R[lc], R[rc]);
}
void Lupdate(int p, int c, int x, int l, int r){
    if(c < l || c > r) return;
    if(l == r){
        L[p] += x;
        return;
    }
    Lupdate(lc, c, x, l, mid);
    Lupdate(rc, c, x, mid + 1, r);
    pull(p);
}
void Rupdate(int p, int c, int x, int l, int r){
    if(c < l || c > r) return;
    if(l == r){
        R[p] += x;
        return;
    }
    Rupdate(lc, c, x, l, mid);
    Rupdate(rc, c, x, mid + 1, r);
    pull(p);
}
int Lquery(int p, int ql, int qr, int l, int r){
    if(ql > r || qr < l) return 2e9;
    if(ql <= l && qr >= r) return L[p];
    return min(Lquery(lc, ql, qr, l, mid), Lquery(rc, ql, qr, mid +
}
int Rquery(int p, int ql, int qr, int l, int r){
    if(ql > r || qr < l) return 2e9;
    if(ql <= l && qr >= r) return R[p];
    return min(Rquery(lc, ql, qr, l, mid), Rquery(rc, ql, qr, mid +
}
signed main(){
    int n, q, k, x, t;
    cin >> n >> q;
    for(int i = 1; i <= n; i++){
        cin >> P[i];
        Lupdate(1, i, i, 1, n);
        Rupdate(1, i, n - i + 1, 1, n);
        Lupdate(1, i, P[i], 1, n);
        Rupdate(1, i, P[i], 1, n);
    }
    while(q--){
        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

```cpp
#include <bits/stdc++.h>
#define int long long
#define mid ((l + r) >> 1)
#define lc (p << 1)
#define rc ((p << 1) | 1)
using namespace std;
array<int, 800004> A, P, M, S;
void pull(int p){
    A[p] = A[lc] + A[rc];
    P[p] = max(P[lc], A[lc] + P[rc]);
    M[p] = max({S[lc] + P[rc], M[lc], M[rc]});
    S[p] = max(S[rc], A[rc] + S[lc]);
}
void update(int p, int c, int x, int l, int r){
    if(c < l || c > r) return;
    if(l == r){
        A[p] = P[p] = M[p] = S[p] = x;
        return;
    }
    update(lc, c, x, l, mid);
    update(rc, c, x, mid + 1, r);
    pull(p);
}
signed main(){
    int n, m, k, x;
    cin >> n >> m;
    for(int i = 1; i <= n; i++){
        cin >> x;
        update(1, i, x, 1, n);
    }
    while(m--){
        cin >> k >> x;
        update(1, k, x, 1, n);
        cout << max(0ll, M[1]) << "\n";
    }
    return 0;
}
```

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

BIT

```cpp
#include <bits/stdc++.h>
#define pb push_back
using namespace std;
struct qq{
    int t, l, r;
};
array<int, 200004> X, BIT, ans;
vector<qq> Q;
map<int, int> P;
bool cmp(qq a, qq b){
    return a.r < b.r;
}
void update(int p, int x){
    for(; p < 200004; p += p & -p){
        BIT[p] += x;
    }
}
int query(int p){
    int sum = 0;
    for(; p > 0; p -= p & -p){
        sum += BIT[p];
    }
    return sum;
}
signed main(){
    int n, q, l, r, p = 0;
    cin >> n >> q;
    for(int i = 1; i <= n; i++){
        cin >> X[i];
    }
    for(int i = 0; i < q; i++){
        cin >> l >> r;
        Q.pb({i, l, r});
    }
    sort(Q.begin(), Q.end(), cmp);
    for(int i = 1; i <= n; i++){
        if(P[X[i]]) update(P[X[i]], -1);
        P[X[i]] = i;
        update(i, 1);
        while(p < Q.size() && Q[p].r == i){
            ans[Q[p].t] = query(Q[p].r) - query(Q[p].l - 1);
            p++;
        }
    }
    for(int i = 0; i < q; i++){
        cout << ans[i] << "\n";
    }
    return 0;
}
```

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

懶標線段樹　二分搜

```cpp
#include <bits/stdc++.h>
#define pb push_back
#define int long long
#define mid ((l + r) >> 1)
#define lc (p << 1)
#define rc ((p << 1) | 1)
using namespace std;
struct qq{
    int l, r, t;
};
bool cmp(qq a, qq b){
    return a.l > b.l;
}
array<int, 200004> x, ans;
array<int, 800004> X, S, M, tag;
vector<qq> Q;
void build(int p, int l, int r){
    if(l == r){
        X[p] = x[l];
        return;
    }
    build(lc, l, mid);
    build(rc, mid + 1, r);
    X[p] = X[lc] + X[rc];
}
void pull(int p){
    S[p] = S[lc] + S[rc];
    M[p] = max(M[lc], M[rc]);
}
void push(int p, int l, int r){
    if(!tag[p]) return;
    S[lc] = (mid - l + 1) * tag[p];
    S[rc] = (r - mid) * tag[p];
    M[lc] = M[rc] = tag[p];
    tag[lc] = tag[rc] = tag[p];
    tag[p] = 0;
}
int find(int p, int v, int l, int r){
    if(l == r) return l;
    push(p, l, r);
    if(M[lc] >= v) return find(lc, v, l, mid);
    else if(M[rc] >= v) return find(rc, v, mid + 1, r);
    else return r +1;
}
void update(int p, int ql, int qr, int v, int l, int r){
    if(ql > r || qr < l) return;
    if(l != r) push(p, l, r);
    if(ql <= l && qr >= r){
        S[p] = (r - l + 1) * v;
        M[p] = v;
        tag[p] = v;
        return;
    }
```

```
    update(lc, ql, qr, v, l, mid);
    update(rc, ql, qr, v, mid + 1, r);
    pull(p);
}
int query(int p, int ql, int qr, int l, int r){
    if(ql > r || qr < l) return 0;
    if(l != r) push(p, l, r);
    if(ql <= l && qr >= r) return S[p] - X[p];
    return query(lc, ql, qr, l, mid) + query(rc, ql, qr, mid + 1, r
}
signed main(){
    int n, q, l, r, p = 0;
    cin >> n >> q;
    for(int i = 1; i <= n; i++){
        cin >> x[i];
    }
    build(1, 1, n);
    for(int i = 0; i < q; i++){
        cin >> l >> r;
        Q.pb({l, r, i});
    }
    sort(Q.begin(), Q.end(), cmp);
    for(int i = n; i > 0; i--){
        update(1, i, find(1, x[i], 1, n) - 1, x[i], 1, n);
        while(Q[p].l == i){
            ans[Q[p].t] = query(1, Q[p].l, Q[p].r, 1, n);
            p++;
        }
    }
    for(int i = 0; i < q; i++){
        cout << ans[i] << "\n";
    }
    return 0;
}
```

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

BIT

```cpp
#include <bits/stdc++.h>
using namespace std;
array<array<int, 1004>, 1004> F, BIT;
void update(int x, int y, int v){
    for(int i = x; i < 1004; i += i & -i){
        for(int j = y; j < 1004; j += j & -j){
            BIT[i][j] += v;
        }
    }
}
int query(int x, int y){
    int sum = 0;
    for(int i = x; i > 0; i -= i & -i){
        for(int j = y; j > 0; j -= j & -j){
            sum += BIT[i][j];
        }
    }
    return sum;
}
signed main(){
    int n, q, x, y, x1, y1, t;
    char f;
    cin >> n >> q;
    for(int i = 1; i <= n; i++){
        for(int j = 1; j <= n; j++){
            cin >> f;
            if(f == '*'){
                update(i, j, 1);
                F[i][j] = 1;
            }else{
                F[i][j] = 0;
            }
        }
    }
    while(q--){
        cin >> t >> x >> y;
        if(t == 1){
            if(F[x][y]) update(x, y, -1);
            else update(x, y, 1);
            F[x][y] ^= 1;
        }else{
            cin >>x1 >> y1;
            cout << query(x1, y1) - query(x1, y - 1) - query(x - 1,
        }
    }
    return 0;
}
```

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

懶標線段樹

```cpp
#include <bits/stdc++.h>
#define int long long
#define mid ((l + r) >> 1)
#define lc (p << 1)
#define rc ((p << 1) | 1)
using namespace std;
array<int, 800004> S, TAG, CHG;
void pull(int p){
    S[p] = S[lc] + S[rc];
}
void push(int p, int l, int r){
    if(CHG[p]){
        TAG[lc] = TAG[rc] = 0;
        CHG[lc] = CHG[rc] = CHG[p];
        S[lc] = CHG[p] * (mid - l + 1);
        S[rc] = CHG[p] * (r - mid);
        CHG[p] = 0;
    }
    TAG[lc] += TAG[p];
    TAG[rc] += TAG[p];
    S[lc] += TAG[p] * (mid - l + 1);
    S[rc] += TAG[p] * (r - mid);
    TAG[p] = 0;
}
void update(int p, int ql, int qr, int x, int l, int r, int t){
    if(ql > r || qr < l) return;
    if(ql <= l && qr >= r){
        if(t){
            S[p] = x * (r - l + 1);
            TAG[p] = 0;
            CHG[p] = x;
        }else{
            S[p] += x * (r - l + 1);
            TAG[p] += x;
        }
        return;
    }
    push(p, l, r);
    update(lc, ql, qr, x, l, mid, t);
    update(rc, ql, qr, x, mid + 1, r, t);
    pull(p);
}
int query(int p, int ql, int qr, int l, int r){
    if(ql > r || qr < l) return 0;
    if(ql <= l && qr >= r) return S[p];
    push(p, l, r);
    return query(lc, ql, qr, l, mid) + query(rc, ql, qr, mid + 1, r
}
signed main(){
    int n, q, l, r, t, x;
    cin >> n >> q;
    for(int i = 1; i <= n; i++){
        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

```cpp
#include <bits/stdc++.h>
#define int long long
#define mid ((l + r) >> 1)
#define lc (p << 1)
#define rc ((p << 1) | 1)
using namespace std;
array<int, 200004> T;
array<int, 800004> S, tag, d;
void pull(int p){
    S[p] = S[lc] + S[rc];
}
void push(int p, int l, int r){
    S[lc] += (mid - l + 1) * (tag[p] + (tag[p] + d[p] * (mid - l)))
    S[rc] += (r - mid) * ((tag[p] + d[p] * (mid + 1 - l)) + (tag[p]
    tag[lc] += tag[p];
    tag[rc] += tag[p] + d[p] * (mid + 1 - l);
    d[lc] += d[p];
    d[rc] += d[p];
    tag[p] = d[p] = 0;
}
void build(int p, int l, int r){
    if(l == r){
        S[p] = T[l];
        return;
    }
    build(lc, l, mid);
    build(rc, mid + 1, r);
    pull(p);
}
void update(int p, int ql, int qr, int v, int l, int r){
    if(ql > r || qr < l) return;
    if(l != r) push(p, l, r);
    if(ql <= l && qr >= r){
        S[p] += (r - l + 1) * (v + (v + (r - l))) / 2ll;
        tag[p] = v;
        d[p] = 1;
        return;
    }
    update(lc, ql, qr, v, l, mid);
    update(rc, ql, qr, v + max(0ll, (mid + 1 - max(ql, l))), mid +
    pull(p);
}
int query(int p, int ql, int qr, int l, int r){
    if(ql > r || qr < l) return 0;
    if(l != r) push(p, l, r);
    if(ql <= l && qr >= r) return S[p];
    return query(lc, ql, qr, l, mid) + query(rc, ql, qr, mid + 1, r
}
signed main(){
    int n, q, t, l, r;
    cin >> n >> q;
    for(int i = 1; i <= n; i++){
        cin >> T[i];
```

```
    }
    build(1, 1, n);
    while(q--){
        cin >> t >> l >> r;
        if(t == 1){
            update(1, l, r, 1, 1, n);
        }else{
            cout << query(1, l, r, 1, n) << "\n";
        }
    }
    return 0;
}
```

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

動態開點持久化線段樹

```cpp
#include <bits/stdc++.h>
#define int long long
#define pb push_back
#define mid ((l + r) >> 1)
using namespace std;
struct seg{
    int val;
    seg *lc, *rc;
    seg(){
        lc = rc = nullptr;
    }
    void pull(){
        val = (lc? lc->val : 0) + (rc? rc->val : 0);
    }
    void build(int l, int r){
        if(l == r) return;
        lc = new seg;
        rc = new seg;
        lc->build(l, mid);
        rc->build(mid + 1, r);
    }
    seg* update(int c, int x, int l, int r){
        seg *root = new seg;
        *root = *this;
        if(c < l || c > r) return root;
        if(l == r){
            root->val = x;
            return root;
        }
        root->lc = lc->update(c, x, l, mid);
        root->rc = rc->update(c, x, mid + 1, r);
        root->pull();
        return root;
    }
    int query(int ql, int qr, int l, int r){
        if(ql > r || qr < l) return 0;
        if(ql <= l && qr >= r) return val;
        return lc->query(ql, qr, l, mid) + rc->query(ql, qr, mid +
    }
};
vector<seg*> S;
signed main(){
    int n, q, t, l, r, x, k;
    cin >> n >> q;
    S.pb(nullptr);
    S.pb(new seg);
    S[1]->build(1, n);
    for(int i = 1; i <= n; i++){
        cin >> x;
        S[1] = S[1]->update(i, x, 1, n);
    }
    while(q--){
        cin >> t >> k;
```

```
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      return 0;
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

```cpp
#include <bits/stdc++.h>
#define pb push_back
using namespace std;
array<int, 200004> in;
array<vector<int>, 200004> T;
int dfs(int u, int pre){
    int cnt = 0;
    for(int v : T[u]){
        if(v == pre) continue;
        cnt += dfs(v, u);
        if(!in[v] && !in[u]){
            cnt++;
            in[u] = 1;
        }
    }
    return cnt;
}
signed main(){
    int n, a, b;
    cin >> n;
    for(int i = 1; i < n; i++){
        cin >> a >> b;
        T[a].pb(b);
        T[b].pb(a);
    }
    cout << dfs(1, 0);
    return 0;
}
```

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

DFS

```cpp
#include <bits/stdc++.h>
#define pb push_back
using namespace std;
array<vector<int>, 200004> T;
int len = 0;
void comp(int &f, int &s, int k){
    if(k >= f) swap(k, f);
    if(k > s) swap(k, s);
}
int dfs(int u, int pre){
    int f = 0, s = 0;
    for(int v : T[u]){
        if(v == pre) continue;
        comp(f, s, dfs(v, u));
    }
    len = max(len, f + s);
    return f + 1;
}
signed main(){
    int n, a, b;
    cin >> n;
    for(int i = 1; i < n; i++){
        cin >> a >> b;
        T[a].pb(b);
        T[b].pb(a);
    }
    dfs(1, 0);
    cout << len;
    return 0;
}
```

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

DFS DP

```cpp
#include <bits/stdc++.h>
#define pb push_back
using namespace std;
array<int, 200004> F, S, D;
array<vector<int>, 200004> T;
void comp(int &f, int &s, int k){
    if(k > f) swap(k, f);
    if(k > s) swap(k, s);
}
int dfs(int u, int pre){
    for(int v : T[u]){
        if(v == pre) continue;
        comp(F[u], S[u], dfs(v, u));
    }
    return F[u] + 1;
}
void dsf(int u, int pre, int len){
    D[u] = max(len, F[u]);
    for(int v : T[u]){
        if(v == pre) continue;
        if(F[u] == F[v] + 1) dsf(v, u, max(S[u], len) + 1);
        else dsf(v, u, max(F[u], len) + 1);
    }
}
signed main(){
    int n, a, b;
    cin >> n;
    for(int i = 1; i < n; i++){
        cin >> a >> b;
        T[a].pb(b);
        T[b].pb(a);
    }
    dfs(1, 0);
    dsf(1, 0, 0);
    for(int i = 1; i <= n; i++){
        cout << D[i] << " ";
    }
    return 0;
}
```

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

DFS DP

```cpp
#include <bits/stdc++.h>
#define pb push_back
#define int long long
using namespace std;
array<int, 200004> sub, sum, ans;
array<vector<int>, 200004> T;
int dfs(int u, int pre){
    sub[u] = 1;
    for(int v : T[u]){
        if(v == pre) continue;
        sum[u] += dfs(v, u);
        sub[u] += sub[v];
    }
    return sum[u] + sub[u];
}
void dsf(int u, int pre, int sm, int sb){
    ans[u] = sum[u] + sm;
    for(int v : T[u]){
        if(v == pre) continue;
        dsf(v, u, sm + sum[u] - sum[v] - sub[v] + sb + sub[u] - sub
    }
}
signed main(){
    int n, a, b;
    cin >> n;
    for(int i = 1; i < n; i++){
        cin >> a >> b;
        T[a].pb(b);
        T[b].pb(a);
    }
    dfs(1, 0);
    dsf(1, 0, 0, 0);
    for(int i = 1; i <= n; i++){
        cout << ans[i] << " ";
    }
    return 0;
}
```

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

Doubling

```cpp
#include <bits/stdc++.h>
using namespace std;
array<array<int, 20>, 200004> B;
void dabo(int n){
    for(int j = 1; j < 20; j++){
        for(int i = 1; i <= n; i++){
            B[i][j] = B[B[i][j - 1]][j - 1];
        }
    }
}
int query(int x, int k){
    for(int i = 19; i >= 0; i--){
        if(k >= 1 << i){
            x = B[x][i];
            k -= 1 << i;
        }
    }
    return x? x : -1;
}
signed main(){
    int n, q, x, k;
    cin >> n >> q;
    for(int i = 2; i <= n; i++){
        cin >> B[i][0];
    }
    dabo(n);
    while(q--){
        cin >> x >> k;
        cout << query(x, k) << "\n";
    }
    return 0;
}
```

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

LCA

```cpp
#include <bits/stdc++.h>
#define pb push_back
using namespace std;
int cnt = 0;
array<int, 200004> in, out;
array<array<int, 20>, 200004> B;
array<vector<int>, 200004> T;
void dfs(int u){
    if(in[u]) return;
    in[u] = ++cnt;
    for(int v : T[u]){
        dfs(v);
    }
    out[u] = ++cnt;
}
void dabo(int n){
    in[0] = 0, out[0] = 1e9;
    for(int j = 1; j < 20; j++){
        for(int i = 1; i <= n; i++){
            B[i][j] = B[B[i][j - 1]][j - 1];
        }
    }
}
int LCA(int a, int b){
    for(int i = 19; i >= 0; i--){
        if(in[B[a][i]] > in[b] || out[B[a][i]] < out[b]){
            a = B[a][i];
        }
    }
    if(in[a] <= in[b] && out[a] >= out[b]) return a;
    return B[a][0];
}
signed main(){
    int n, q, a, b;
    cin >> n >> q;
    for(int i = 2; i <= n; i++){
        cin >> B[i][0];
        T[B[i][0]].pb(i);
    }
    dfs(1);
    dabo(n);
    while(q--){
        cin >> a >> b;
        cout << LCA(a, b) << "\n";
    }
    return 0;
}
```

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

LCA

```cpp
#include <bits/stdc++.h>
#define pb push_back
using namespace std;
int cnt = 0;
array<int, 200004> in, out;
array<array<int , 20>, 200004> A;
array<vector<int>, 200004> T;
void dfs(int u){
    in[u] = ++cnt;
    for(int v : T[u]){
        if(in[v]) continue;
        dfs(v);
        A[v][0] = u;
    }
    out[u] = ++cnt;
}
void dabo(int n){
    in[0] = 0, out[0] = 1e9;
    for(int j = 1; j < 20; j++){
        for(int i = 1; i <= n; i++){
            A[i][j] = A[A[i][j - 1]][j - 1];
        }
    }
}
int LCA(int a, int b){
    int dis = 0;
    for(int i = 19; i >= 0; i--){
        if(in[A[a][i]] >= in[b] || out[A[a][i]] <= out[b]){
            dis += 1 << i;
            a = A[a][i];
        }
    }
    if(in[a] > in[b] || out[a] < out[b]){
        dis++;
        a = A[a][0];
    }
    for(int i = 19; i >= 0; i--){
        if(in[A[b][i]] >= in[a] || out[A[b][i]] <= out[a]){
            dis += 1 << i;
            b = A[b][i];
        }
    }
    return dis;
}
signed main(){
    int n, q, a, b;
    cin >> n >> q;
    for(int i = 1; i < n; i++){
        cin >> a >> b;
        T[a].pb(b);
        T[b].pb(a);
    }
    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

```cpp
#include <bits/stdc++.h>
#define pb push_back
using namespace std;
int cnt = 0;
array<int, 200004> ans, P, in, out;
array<array<int, 20>, 200004> A;
array<vector<int>, 200004> T;
void dfs(int u){
    in[u] = ++cnt;
    for(int v : T[u]){
        if(in[v]) continue;
        dfs(v);
        A[v][0] = u;
    }
    out[u] = ++cnt;
}
void dabo(int n){
    in[0] = 0, out[0] = 1e9;
    for(int j = 1; j < 20; j++){
        for(int i = 1; i <= n; i++){
            A[i][j] = A[A[i][j - 1]][j - 1];
        }
    }
}
int LCA(int a, int b){
    for(int i = 19; i >= 0; i--){
        if(in[A[a][i]] > in[b] || out[A[a][i]] < out[b]) a = A[a][i
    }
    if(in[a] > in[b] || out[a] < out[b]) a = A[a][0];
    return a;
}
int dsf(int u, int pre){
    for(int v : T[u]){
        if(v == pre) continue;
        P[u] += dsf(v, u);
    }
    return P[u];
}
signed main(){
    int n, q, a, b, c;
    cin >> n >> q;
    for(int i = 1; i < n; i++){
        cin >> a >> b;
        T[a].pb(b);
        T[b].pb(a);
    }
    dfs(1);
    dabo(n);
    while(q--){
        cin >> a >> b;
        c = LCA(a, b);
        P[a]++;
        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

```cpp
#include <bits/stdc++.h>
#define int long long
#define pb push_back
using namespace std;
int cnt = 0;
array<int, 200004> BIT, L, P, V;
array<vector<int>, 200004> T;
void update(int p, int v){
    for(; p < 200004; p += p & -p) BIT[p] += v;
}
int query(int p){
    int sum = 0;
    for(; p > 0; p -= p & -p) sum += BIT[p];
    return sum;
}
int press(int u, int pre){
    L[u] = 1e9;
    for(int v : T[u]){
        if(v == pre) continue;
        L[u] = min(L[u], press(v, u));
    }
    P[u] = ++cnt;
    return L[u] = min(L[u], P[u]);
}
signed main(){
    int n, q, a, b, t, s, x;
    cin >> n >> q;
    for(int i = 1; i <= n; i++){
        cin >> V[i];
    }
    for(int i = 1; i < n; i++){
        cin >> a >> b;
        T[a].pb(b);
        T[b].pb(a);
    }
    press(1, 0);
    for(int i = 1; i <= n; i++){
        update(P[i], V[i]);
    }
    while(q--){
        cin >> t >> s;
        if(t == 1){
            cin >> x;
            update(P[s], x - V[s]);
            V[s] = x;
        }else cout << query(P[s]) - query(L[s] - 1) << "\n";
    }
    return 0;
}
```

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

樹鍊剖分　BIT

```cpp
#include <bits/stdc++.h>
#define int long long
#define pb push_back
using namespace std;
struct BIT{
    vector<int> bit;
    void update(int p, int v){
        for(; p < bit.size(); p += p & -p) bit[p] += v;
    }
    int query(int p){
        int sum = 0;
        for(; p > 0; p -= p & -p) sum += bit[p];
        return sum;
    }
};
int cnt = 1;
array<int, 200004> V, H, P, S, D, pre, C;
array<vector<int>, 200004> T;
array<BIT, 200004> B;
int dfsiz(int u, int p, int dep){
    int tmp, siz = 1, mx = 0;
    D[u] = dep;
    pre[u] = p;
    for(int v : T[u]){
        if(v == p) continue;
        tmp = dfsiz(v, u, dep + 1);
        siz += tmp;
        if(tmp > mx){
            mx = tmp;
            S[u] = v;
        }
    }
    return siz;
}
void cut(int u, int h, int c, int p){
    H[u] = h;
    C[u] = c;
    P[u] = p;
    if(p == 1) B[c].bit.pb(0);
    B[c].bit.pb(0);
    for(int v : T[u]){
        if(v == pre[u]) continue;
        if(v == S[u]) cut(v, h, c, p + 1);
        else cut(v, v, ++cnt, 1);
    }
}
int path(int s){
    int ans = 0;
    while(s){
        ans += B[C[s]].query(P[s]);
        s = pre[H[s]];
    }
    return ans;
```

```
}
signed main(){
    int n, q, a, b, t, s, x;
    cin >> n >> q;
    for(int i = 1; i <= n; i++){
        cin >> V[i];
    }
    for(int i = 1; i < n; i++){
        cin >> a >> b;
        T[a].pb(b);
        T[b].pb(a);
    }
    dfsiz(1, 0, 1);
    cut(1, 1, 1, 1);
    for(int i = 1; i <= n; i++){
        B[C[i]].update(P[i], V[i]);
    }
    while(q--){
        cin >> t >> s;
        if(t == 1){
            cin >> x;
            B[C[s]].update(P[s], x - V[s]);
            V[s] = x;
        }else cout << path(s) << "\n";
    }
    return 0;
}
```

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

樹鍊剖分　線段樹

```cpp
#include <bits/stdc++.h>
#define mid ((l + r) >> 1)
#define pb push_back
using namespace std;
struct seg{
    int val;
    seg *lc, *rc;
    seg(){val = 0; lc = rc = nullptr;}
    void pull(){
        val = max(lc? lc->val : 0, rc? rc->val : 0);
    }
    void update(int x, int v, int l, int r){
        if(l == r){
            val = v;
            return;
        }
        if(x <= mid){
            if(!lc) lc = new seg;
            lc->update(x, v, l, mid);
        }else{
            if(!rc) rc = new seg;
            rc->update(x, v, mid + 1, r);
        }
        pull();
    }
    int query(int ql, int qr, int l, int r){
        if(ql > r || qr < l) return 0;
        if(ql <= l && qr >= r) return val;
        return max(lc? lc->query(ql, qr, l, mid) : 0, rc? rc->quer
    }
};
int cnt = 1, n;
array<int, 200004> P, C, D, pre, H, M, V, Z;
array<vector<int>, 200004> T;
array<seg*, 200004> S;
int dfs(int u, int p, int dep){
    int s = 1, tmp, mx = 0;
    D[u] = dep;
    pre[u] = p;
    for(int v : T[u]){
        if(v == p) continue;
        tmp = dfs(v, u, dep + 1);
        s += tmp;
        if(tmp > mx){
            mx = tmp;
            M[u] = v;
        }
    }
    return s;
}
void cut(int u, int h, int c, int p){
    H[u] = h;
    C[u] = c;
```

```
    P[u] = p;
    Z[c] = p;
    for(int v : T[u]){
        if(v == pre[u]) continue;
        if(v == M[u]) cut(v, h, c, p + 1);
        else cut(v, v, ++cnt, 1);
    }
}
int path(int a, int b){
    int ans = 0;
    while(H[a] != H[b]){
        if(D[H[a]] < D[H[b]]) swap(a, b);
        ans = max(ans, S[C[a]]->query(1, P[a], 1, Z[C[a]]));
        a = pre[H[a]];
    }
    if(D[a] < D[b]) swap(a, b);
    return max(ans, S[C[a]]->query(P[b], P[a], 1, Z[C[a]]));
}
signed main(){
    cin.tie(0), cout.tie(0), ios::sync_with_stdio(0);
    int q, a, b, t, s, x;
    cin >> n >> q;
    for(int i = 1; i <= n; i++){
        cin >> V[i];
    }
    for(int i = 1; i < n; i++){
        cin >> a >> b;
        T[a].pb(b);
        T[b].pb(a);
    }
    dfs(1, 0, 1);
    cut(1, 1, 1, 1);
    for(int i = 1; i <= n; i++){
        if(!S[C[i]]) S[C[i]] = new seg;
        S[C[i]]->update(P[i], V[i], 1, Z[C[i]]);
    }
    while(q--){
        cin >> t;
        if(t == 1){
            cin >> s >> x;
            S[C[s]]->update(P[s], x, 1, Z[C[s]]);
        }else{
            cin >> a >> b;
            cout << path(a, b) << "\n";
        }
    }
    return 0;
}
```

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

Set 啟發式合併

```cpp
#include <bits/stdc++.h>
#define pb push_back
#define ins insert
using namespace std;
array<int, 200004> C, ans;
array<vector<int>, 200004> T;
set<int> dfs(int u, int pre){
    set<int> S, tmp;
    S.ins(C[u]);
    for(int v : T[u]){
        if(v == pre) continue;
        tmp = dfs(v, u);
        if(tmp.size() > S.size()) swap(S, tmp);
        for(int it : tmp){
            S.ins(it);
        }
    }
    ans[u] = S.size();
    return S;
}
signed main(){
    int n, a, b;
    cin >> n;
    for(int i = 1; i <= n; i++){
        cin >> C[i];
    }
    for(int i = 1; i < n; i++){
        cin >> a >> b;
        T[a].pb(b);
        T[b].pb(a);
    }
    dfs(1, 0);
    for(int i = 1; i <= n; i++){
        cout << ans[i] << " ";
    }
    return 0;
}
```

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

DFS

```cpp
#include <bits/stdc++.h>
#define pb push_back
using namespace std;
int n, cen;
array<vector<int>, 200004> T;
int dfs(int u, int pre){
    int s = 1, mux = 0, tmp;
    for(int v : T[u]){
        if(v == pre) continue;
        tmp = dfs(v, u);
        s += tmp;
        mux = max(mux, tmp);
    }
    if(max(mux, n - s) <= n / 2) cen = u;
    return s;
}
signed main(){
    int a, b;
    cin >> n;
    for(int i = 1; i < n; i++){
        cin >> a >> b;
        T[a].pb(b);
        T[b].pb(a);
    }
    dfs(1, 0);
    cout << cen;
    return 0;
}
```

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

重心剖分

```cpp
#include <bits/stdc++.h>
#define int long long
#define pb push_back
using namespace std;
int n, k;
array<bool, 200004> vis;
array<int, 200004> S, M, cnt;
array<vector<int>, 200004> T, C;
vector<int> leaf;
int dfsiz(int u){
    if(vis[u]) return 0;
    int tmp;
    leaf.pb(u);
    vis[u] = 1;
    S[u] = 1;
    M[u] = 0;
    for(int v : T[u]){
        tmp = dfsiz(v);
        M[u] = max(M[u], tmp);
        S[u] += tmp;
    }
    return S[u];
}
int cut(int root){
    leaf.clear();
    int cen, s;
    dfsiz(root);
    s = leaf.size();
    for(int u : leaf){
        if(max(M[u], s - S[u]) <= s / 2) cen = u;
        vis[u] = 0;
    }
    vis[cen] = 1;
    for(int v : T[cen]){
        if(!vis[v]) C[cen].pb(cut(v));
    }
    return cen;
}
int dfs(int u, int pre, int s){
    if(vis[u] || s > k) return 0;
    int sum = 0;
    sum += cnt[k - s];
    cnt[s]++;
    for(int v : T[u]){
        if(v == pre) continue;
        sum += dfs(v, u, s + 1);
    }
    return sum;
}
int path(int u){
    int ans = 0, tmp;
    ans += dfs(u, u, 0);
    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)

重心剖分　BIT

```cpp
#include <bits/stdc++.h>
#define int long long
#define pb push_back
using namespace std;
int n, k1, k2, d;
array<bool, 200004> vis;
array<int, 200004> S, M, BNT;
array<vector<int>, 200004> T, C;
vector<int> leaf, see;
void update(int p){
    p++;
    for(; p <= n; p += p & -p){
        if(!BNT[p]) see.pb(p);
        BNT[p]++;
    }
}
int query(int p){
    p++;
    if(p <= 0) return 0;
    int sum = 0;
    for(; p > 0; p -= p & -p) sum += BNT[p];
    return sum;
}
int dfsiz(int u){
    if(vis[u]) return 0;
    int tmp;
    leaf.pb(u);
    vis[u] = 1;
    S[u] = 1;
    M[u] = 0;
    for(int v : T[u]){
        tmp = dfsiz(v);
        S[u] += tmp;
        M[u] = max(M[u], tmp);
    }
    return S[u];
}
int dfs(int u, int pre, int s){
    if(vis[u] || s > k2) return 0;
    int sum = 0;
    sum += query(k2 - s) - query(k1 - s - 1);
    update(s);
    for(int v : T[u]){
        if(v == pre) continue;
        sum += dfs(v, u, s + 1);
    }
    return sum;
}
int cut(int root){
    int cen, s, ans = 0;
    leaf.clear();
    dfsiz(root);
    s = leaf.size();
```

```
    for(int u : leaf){
        if(max(M[u], s - S[u]) <= s / 2) cen = u;
        vis[u] = 0;
    }
    ans += dfs(cen, cen, 0);
    for(int s : see) BNT[s] = 0;
    see.clear();
    vis[cen] = 1;
    for(int v : T[cen]){
        ans -= dfs(v, cen, 1);
        for(int s : see) BNT[s] = 0;
        see.clear();
    }
    for(int v : T[cen]){
        if(!vis[v]) ans += cut(v);
    }
    return ans;
}
signed main(){
    cin.tie(0), cout.tie(0), ios::sync_with_stdio(0);
    int a, b;
    cin >> n >> k1 >> k2;
    for(int i = 1; i < n; i++){
        cin >> a >> b;
        T[a].pb(b);
        T[b].pb(a);
    }
    cout << cut(1);
    return 0;
}
```

Mathematics

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

遞迴

```cpp
#include <bits/stdc++.h>
using namespace std;
int jos(int n, int k, int f){
    if(n == 1) return 1;
    int kill = (n + f) / 2, nk;
    if(k <= kill) return 2 * k - f;
    else{
        nk = jos(n - kill, k - kill, (n ^ f) & 1);
        return 2 * nk - (1 ^ f);
    }
}
signed main(){
    int q, n, k;
    cin >> q;
    while(q--){
        cin >> n >> k;
        cout << jos(n, k, 0) << "\n";
    }
    return 0;
}
```

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

快速冪

```cpp
#include <bits/stdc++.h>
#define int long long
using namespace std;
const int mod = 1e9 + 7;
int power(int x, int k){
    int ans = 1;
    for(int i = 1; i <= k; i <<= 1){
        if(k & i) ans *= x, ans %= mod;
        x *= x, x %= mod;
    }
    return ans;
}
signed main(){
    int n, a, b;
    cin >> n;
    while(n--){
        cin >> a >> b;
        cout << power(a, b) << "\n";
    }
    return 0;
}
```

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

快速冪　費馬小定理

```cpp
#include <bits/stdc++.h>
#define int long long
using namespace std;
int power(int x, int k, int p){
    int ans = 1;
    for(int i = 1; i <= k; i <<= 1){
        if(i & k) ans *= x, ans %= p;
        x *= x, x %= p;
    }
    return ans;
}
signed main(){
    int n, a, b, c, bc, p = 1e9 + 7;
    cin >> n;
    while(n--){
        cin >> a >> b >> c;
        bc = power(b, c, p - 1);
        cout << power(a, bc, p) << "\n";
    }
    return 0;
}
```

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

根號

```cpp
#include <bits/stdc++.h>
using namespace std;
int div(int x){
    int cnt = 0;
    for(int i = 1; i * i <= x; i++){
        if(x % i == 0) cnt += 2;
        if(i * i == x) cnt--;
    }
    return cnt;
}
signed main(){
    int n, x;
    cin >> n;
    while(n--){
        cin >> x;
        cout << div(x) << "\n";
    }
    return 0;
}
```

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

類質數篩

```cpp
#include <bits/stdc++.h>
using namespace std;
array<int, 1000004> cnt;
int gcd(){
    int div;
    for(int i = 1e6; i > 0; i--){
        div = 0;
        for(int j = i; j <= 1e6; j += i){
            div += cnt[j];
        }
        if(div > 1) return i;
    }
}
signed main(){
    int n, x;
    cin >> n;
    while(n--){
        cin >> x;
        cnt[x]++;
    }
    cout << gcd();
    return 0;
}
```

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

根號　模逆元

```cpp
#include <bits/stdc++.h>
#define int long long
using namespace std;
const int mod = 1e9 + 7, m = 5e8 + 4;
signed main(){
    int n, ans = 0, l, r, gh;
    cin >> n;
    gh = sqrt(n);
    for(int i = 1; i <= gh; i++){
        l = n / i, r = n / (i + 1) + 1;
        ans += (((((((l - r + 1) % mod) * ((l + r) % mod)) % mod) *
        ans %= mod;
    }
    for(int i = 1; i <= n / (gh + 1); i++){
        ans += (n / i) * i;
        ans %= mod;
    }
    cout << ans;
    return 0;
}
```

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

快速冪　模逆元

```cpp
#include <bits/stdc++.h>
#define int long long
using namespace std;
const int mod = 1e9 + 7;
int exp(int x, int k){
    int ans = 1;
    for(int i = 1; i <= k; i <<= 1){
        if(i & k) ans *= x, ans %= mod;
        x *= x, x %= mod;
    }
    return ans;
}
signed main(){
    int n, x, k, num = 1, sum = 1, pro = 1, cnt = 1;
    cin >> n;
    for(int i = 0; i < n; i++){
        cin >> x >> k;
        num *= (k + 1), num %= mod;
        sum *= (((exp(x, k + 1) - 1 + mod) % mod) * exp(x - 1, mod
        pro = (exp(pro, k + 1) * exp(exp(x, k * (k + 1) / 2), cnt))
        cnt *= (k + 1), cnt %= (mod - 1);
    }
    cout << num << " " << sum << " " << pro;
    return 0;
}
```

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

位元枚舉　排容

```cpp
#include <bits/stdc++.h>
#define int long long
using namespace std;
array<int, 20> A;
int mul(int n, int k){
    int ans = 0, tmp, cnt;
    for(int i = 1; i < 1 << k; i++){
        tmp = n, cnt = 0;
        for(int j = 0; j < k; j++){
            if(i & (1 << j)){
                tmp /= A[j];
                cnt++;
            }
        }
        if(cnt & 1) ans += tmp;
        else ans -= tmp;
    }
    return ans;
}
signed main(){
    int n, k;
    cin >> n >> k;
    for(int i = 0; i < k; i++){
        cin >> A[i];
    }
    cout << mul(n, k);
    return 0;
}
```

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

因數分解　質數篩

```cpp
#include <bits/stdc++.h>
#define int long long
#define pb push_back
using namespace std;
array<int, 1000004> cnt;
array<bool, 1000004> pri;
vector<int> P, C;
void colander(int n){
    pri[1] = 1;
    for(int i = 2; i <= n; i++){
        if(pri[i]) continue;
        for(int j = 2 * i; j <= n; j += i){
            pri[j] = 1;
        }
    }
}
void dfs(int p, int x){
    if(p >= P.size()){
        cnt[x]++;
        return;
    }
    int tmp = 1;
    for(int i = 0; i <= C[p]; i++){
        dfs(p + 1, x * tmp);
        tmp *= P[p];
    }
}
void div(int x){
    P.clear();
    C.clear();
    if(!pri[x]){
        cnt[x]++;
        cnt[1]++;
        return;
    }
    for(int i = 2; i * i <= x; i++){
        if(x % i == 0){
            P.pb(i);
            C.pb(1);
            x /= i;
        }
        while(x % i == 0){
            C[C.size() - 1]++;
            x /= i;
        }
    }
    if(x > 1) P.pb(x), C.pb(1);
    dfs(0, 1);
}
void gcd(int i){
    cnt[i] = cnt[i] * (cnt[i] - 1) / 2;
    for(int j = 2 * i; j <= 1e6; j += i) cnt[i] -= cnt[j];
}
```

```
signed main(){
    cin.tie(0), cout.tie(0), ios::sync_with_stdio(0);
    int n, x;
    cin >> n;
    colander(1e6);
    for(int i = 0; i < n; i++){
        cin >> x;
        div(x);
    }
    for(int i = 1e6; i > 0; i--){
        gcd(i);
    }
    cout << cnt[1];
    return 0;
}
```

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

組合

```cpp
#include <bits/stdc++.h>
#define int long long
using namespace std;
const int mod = 1e9 + 7;
array<int, 1000004> fac;
void fact(int n){
    fac[0] = 1;
    for(int i = 1; i <= n; i++){
        fac[i] = (fac[i - 1] * i) % mod;
    }
}
int exp(int x, int k){
    int ans = 1;
    for(int i = 1; i <= k; i <<= 1){
        if(i & k) ans *= x, ans %= mod;
        x *= x, x %= mod;
    }
    return ans;
}
int C(int n, int k){
    return (((fac[n] * exp(fac[k], mod - 2)) % mod) * exp(fac[n - k
}
signed main(){
    int n, a, b;
    cin >> n;
    fact(1e6);
    while(n--){
        cin >> a >> b;
        cout << C(a, b) << "\n";
    }
    return 0;
}
```

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

排列

```cpp
#include <bits/stdc++.h>
#define int long long
using namespace std;
int mod = 1e9 + 7;
array<int, 1000004> fac;
array<int, 26> cnt;
void fact(int n){
    fac[0] = 1;
    for(int i = 1; i <= n; i++){
        fac[i] = (fac[i - 1] * i) % mod;
    }
}
int exp(int x, int k){
    int ans = 1;
    for(int i = 1; i <= k; i <<= 1){
        if(i & k) ans = (ans * x) % mod;
        x = (x * x) % mod;
    }
    return ans;
}
signed main(){
    int ans;
    string S;
    cin >> S;
    fact(1e6);
    for(char s : S){
        cnt[s - 'a']++;
    }
    ans = fac[S.size()];
    for(int i = 0; i < 26; i++){
        ans = (ans * exp(fac[cnt[i]], mod - 2)) % mod;
    }
    cout << ans;
    return 0;
}
```

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

組合

```cpp
#include <bits/stdc++.h>
#define int long long
using namespace std;
const int mod = 1e9 + 7;
array<int, 2000004> fac;
void fact(int n){
    fac[0] = 1;
    for(int i = 1; i <= n; i++){
        fac[i] = (fac[i - 1] * i) % mod;
    }
}
int exp(int x, int k){
    int ans = 1;
    for(int i = 1; i <= k; i <<= 1){
        if(i & k) ans = (ans * x) % mod;
        x = (x * x) % mod;
    }
    return ans;
}
int C(int n, int k){
    return (((fac[n] * exp(fac[k], mod - 2)) % mod) * exp(fac[n - k
}
signed main(){
    int n, m;
    fact(2e6);
    cin >> n >> m;
    cout << C(m + n - 1, n - 1);
    return 0;
}
```

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

組合　排容

```cpp
#include <bits/stdc++.h>
#define int long long
using namespace std;
const int mod = 1e9 + 7;
array<int, 1000004> fac;
void fact(int n){
    fac[0] = 1;
    for(int i = 1; i <= n; i++){
        fac[i] = (fac[i - 1] * i) % mod;
    }
}
int exp(int x, int k){
    int ans = 1;
    for(int i = 1; i <= k; i <<= 1){
        if(i & k) ans = (ans * x) % mod;
        x = (x * x) % mod;
    }
    return ans;
}
int C(int n, int k){
    return (((fac[n] * exp(fac[k], mod - 2)) % mod) * exp(fac[n - k
}
signed main(){
    int n, ans = 0;
    fact(1e6);
    cin >> n;
    for(int i = n; i >= 0; i--){
        if((n - i) & 1) ans -= (C(n, i) * fac[i]) % mod;
        else ans += (C(n, i) * fac[i]) % mod;
        ans = (ans + mod) % mod;
    }
    cout << ans;
    return 0;
}
```

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

卡特蘭數

```cpp
#include <bits/stdc++.h>
#define int long long
using namespace std;
const int mod = 1e9 + 7;
array<int, 1000004> fac;
void fact(int n){
    fac[0] = 1;
    for(int i = 1; i <= n; i++){
        fac[i] = (fac[i - 1] * i) % mod;
    }
}
int exp(int x, int k){
    int ans = 1;
    for(int i = 1; i <= k; i <<= 1){
        if(i & k) ans = (ans * x) % mod;
        x = (x * x) % mod;
    }
    return ans;
}
int C(int n, int k){
    return (((fac[n] * exp(fac[k], mod - 2)) % mod) * exp(fac[n - k
}
signed main(){
    int n;
    fact(1e6);
    cin >> n;
    if(n & 1) cout << 0;
    else cout << (C(n, n / 2) * exp(n / 2 + 1, mod - 2)) % mod;
    return 0;
}
```

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

卡特蘭數

```cpp
#include <bits/stdc++.h>
#define int long long
using namespace std;
const int mod = 1e9 + 7;
array<int, 1000004> fac;
void fact(int n){
    fac[0] = 1;
    for(int i = 1; i <= n; i++){
        fac[i] = (fac[i - 1] * i) % mod;
    }
}
int exp(int x, int k){
    int ans = 1;
    for(int i = 1; i <= k; i <<= 1){
        if(i & k) ans = (ans * x) % mod;
        x = (x * x) % mod;
    }
    return ans;
}
signed main(){
    int n, l = 0, r = 0, ans;
    string S;
    cin >> n >> S;
    fact(1e6);
    if(n & 1){
        cout << 0;
        return 0;
    }
    n /= 2;
    for(char s : S){
        if(s == '(') l++;
        if(s == ')') r++;
        if(r > l){
            cout << 0;
            return 0;
        }
    }
    if(l > n){
        cout << 0;
        return 0;
    }
    ans = (((fac[2 * n - l - r] * exp(fac[n - l], mod - 2)) % mod)
    cout << (ans - (((fac[2 * n - l - r] * exp(fac[n - r + 1], mod
    return 0;
}
```

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

Burn Side Lemma

```cpp
#include <bits/stdc++.h>
#define int long long
using namespace std;
const int mod = 1e9 + 7;
int gcd(int a, int b){
    return b? gcd(b, a % b) : a;
}
int exp(int x, int k){
    int ans = 1;
    for(int i = 1; i <= k; i <<= 1){
        if(i & k) ans = (ans * x) % mod;
        x = (x * x) % mod;
    }
    return ans;
}
signed main(){
    int n, m, ans = 0;
    cin >> n >> m;
    for(int i = 0; i < n; i++){
        ans = (ans + exp(m, gcd(n, i))) % mod;
    }
    cout << (ans * exp(n, mod - 2)) % mod;
    return 0;
}
```

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

Burn Side Lemma

```cpp
#include <bits/stdc++.h>
#define int long long
using namespace std;
const int mod = 1e9 + 7;
int exp(int x, int k){
    int ans = 1;
    for(int i = 1; i <= k; i <<= 1){
        if(i & k) ans = (ans * x) % mod;
        x = (x * x) % mod;
    }
    return ans;
}
signed main(){
    int n, n0, n1, n2, ans = 0;
    cin >> n;
    if(n == 1){
        cout << 2;
        return 0;
    }
    n0 = n * n;
    if(n & 1){
        n1 = (n * n - 1) / 4 + 1;
        n2 = (n * n - 1) / 2 + 1;
    }else{
        n1 = n * n / 4;
        n2 = n * n / 2;
    }
    ans += exp(2, n0);
    ans += exp(2, n1 + 1);
    ans += exp(2, n2);
    ans %= mod;
    ans = (ans * exp(4, mod - 2)) % mod;
    cout << ans;
    return 0;
}
```

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

矩陣快速冪

```cpp
#include <bits/stdc++.h>
#define int long long
#define matrix array<array<int, 2>, 2>
using namespace std;
const int mod = 1e9 + 7;
matrix K;
matrix mul(matrix A, matrix B){
    matrix C;
    for(int i = 0; i < 2; i++){
        for(int j = 0; j < 2; j++){
            C[i][j] = 0;
            for(int k = 0; k < 2; k++){
                C[i][j] += (A[i][k] * B[k][j]) % mod;
            }
            C[i][j] %= mod;
        }
    }
    return C;
}
int fib(int n){
    matrix F;
    K[0][0] = 1, K[0][1] = 1, K[1][0] = 1, K[1][1] = 0;
    F[0][0] = 1, F[0][1] = 0, F[1][0] = 0, F[1][1] = 1;
    for(int i = 1; i <= n; i <<= 1){
        if(i & n) F = mul(K, F);
        K = mul(K, K);
    }
    return F[0][1];
}
signed main(){
    int n;
    cin >> n;
    cout << fib(n);
    return 0;
}
```

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

矩陣快速冪

```cpp
#include <bits/stdc++.h>
#define int long long
#define matrix array<array<int, 6>, 6>
using namespace std;
const int mod = 1e9 + 7;
matrix K;
matrix mul(matrix A, matrix B){
    matrix C;
    for(int i = 0; i < 6; i++){
        for(int j = 0; j < 6; j++){
            C[i][j] = 0;
            for(int k = 0; k < 6; k++){
                C[i][j] += (A[i][k] * B[k][j]) % mod;
            }
            C[i][j] %= mod;
        }
    }
    return C;
}
int dice(int n){
    matrix D;
    D[0][0] = 1, D[0][1] = 0, D[0][2] = 0, D[0][3] = 0, D[0][4] = 0
    D[1][0] = 0, D[1][1] = 1, D[1][2] = 0, D[1][3] = 0, D[1][4] = 0
    D[2][0] = 0, D[2][1] = 0, D[2][2] = 1, D[2][3] = 0, D[2][4] = 0
    D[3][0] = 0, D[3][1] = 0, D[3][2] = 0, D[3][3] = 1, D[3][4] = 0
    D[4][0] = 0, D[4][1] = 0, D[4][2] = 0, D[4][3] = 0, D[4][4] = 1
    D[5][0] = 0, D[5][1] = 0, D[5][2] = 0, D[5][3] = 0, D[5][4] = 0
    K[0][0] = 1, K[0][1] = 1, K[0][2] = 1, K[0][3] = 1, K[0][4] = 1
    K[1][0] = 1, K[1][1] = 0, K[1][2] = 0, K[1][3] = 0, K[1][4] = 0
    K[2][0] = 0, K[2][1] = 1, K[2][2] = 0, K[2][3] = 0, K[2][4] = 0
    K[3][0] = 0, K[3][1] = 0, K[3][2] = 1, K[3][3] = 0, K[3][4] = 0
    K[4][0] = 0, K[4][1] = 0, K[4][2] = 0, K[4][3] = 1, K[4][4] = 0
    K[5][0] = 0, K[5][1] = 0, K[5][2] = 0, K[5][3] = 0, K[5][4] = 1
    for(int i = 1; i <= n; i <<= 1){
        if(i & n) D = mul(K, D);
        K = mul(K, K);
    }
    return D[0][0];
}
signed main(){
    int n;
    cin >> n;
    cout << dice(n);
    return 0;
}
```

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

矩陣快速冪

```cpp
#include <bits/stdc++.h>
#define int long long
#define matrix array<array<int, 101>, 101>
using namespace std;
const int mod = 1e9 + 7;
matrix G;
matrix mul(matrix A, matrix B, int n){
    matrix C;
    for(int i = 1; i <= n; i++){
        for(int j = 1; j <= n; j++){
            C[i][j] = 0;
            for(int k = 1; k <= n; k++){
                C[i][j] += (A[i][k] * B[k][j]) % mod;
            }
            C[i][j] %= mod;
        }
    }
    return C;
}
int walk(int n, int k){
    matrix W;
    for(int i = 1; i <= n; i++){
        for(int j = 1; j <= n; j++){
            W[i][j] = 0;
        }
        W[i][i] = 1;
    }
    for(int i = 1; i <= k; i <<= 1){
        if(i & k) W = mul(G, W, n);
        G = mul(G, G, n);
    }
    return W[n][1];
}
signed main(){
    int n, m, k, a, b;
    cin >> n >> m >> k;
    while(m--){
        cin >> a >> b;
        G[b][a]++;
    }
    cout << walk(n, k);
    return 0;
}
```

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

矩陣

```cpp
#include <bits/stdc++.h>
#define int long long
#define matrix array<array<int, 101>, 101>
using namespace std;
matrix G, W;
int min(int a, int b){
    if(a < 0) return b;
    if(b < 0) return a;
    return a < b? a : b;
}
matrix mul(matrix A, matrix B, int n){
    matrix C;
    for(int i = 1; i <= n; i++){
        for(int j = 1; j <= n; j++){
            C[i][j] = -1;
            for(int k = 1; k <= n; k++){
                if(A[i][k] < 0 || B[k][j] < 0) continue;
                C[i][j] = min(C[i][j], A[i][k] + B[k][j]);
            }
        }
    }
    return C;
}
int walk(int n, int k){
    k--;
    for(int i = 1; i <= k; i <<= 1){
        if(i & k) W = mul(G, W, n);
        G = mul(G, G, n);
    }
    return W[n][1];
}
signed main(){
    int n, m, k, a, b, c;
    cin >> n >> m >> k;
    for(int i = 1; i <= n; i++){
        for(int j = 1; j <= n; j++){
            G[i][j] = -1;
            W[i][j] = -1;
        }
    }
    while(m--){
        cin >> a >> b >> c;
        G[b][a] = min(G[b][a], c);
        W[b][a] = min(W[b][a], c);
    }
    cout << walk(n, k);
    return 0;
}
```

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

機率 DP

```cpp
#include <bits/stdc++.h>
using namespace std;
array<array<double, 604>, 104> dp;
signed main(){
    int n, a, b;
    double all = 0, ans = 0;
    cin >> n >> a >> b;
    dp[0][0] = 1;
    for(int i = 1; i <= n; i++){
        for(int j = i; j <= 6 * i; j++){
            for(int k = 1; k <= 6; k++){
                if(k > j) break;
                dp[i][j] += dp[i - 1][j - k];
            }
        }
    }
    for(int i = 1; i <= 6 * n; i++){
        all += dp[n][i];
        if(i >= a && i <= b) ans += dp[n][i];
    }
    cout << fixed << setprecision(6) << ans / all;
    return 0;
}
```

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

機率 DP

```cpp
#include <bits/stdc++.h>
using namespace std;
array<array<double, 8>, 8> cnt, dp, tmp, E;
void build(){
    for(int i = 0; i < 8; i++){
        for(int j = 0; j < 8; j++){
            E[i][j] = 1;
            if(i != 0) cnt[i][j] += 1;
            if(i != 7) cnt[i][j] += 1;
            if(j != 0) cnt[i][j] += 1;
            if(j != 7) cnt[i][j] += 1;
        }
    }
}
void reset(){
    for(int i = 0; i < 8; i++){
        for(int j = 0; j < 8; j++){
            dp[i][j] = 0;
        }
    }
}
void clear(){
    for(int i = 0; i < 8; i++){
        for(int j = 0; j < 8; j++){
            tmp[i][j] = 0;
        }
    }
}
void move(int i, int j){
    if(i != 0) tmp[i - 1][j] += dp[i][j] / cnt[i][j];
    if(i != 7) tmp[i + 1][j] += dp[i][j] / cnt[i][j];
    if(j != 0) tmp[i][j - 1] += dp[i][j] / cnt[i][j];
    if(j != 7) tmp[i][j + 1] += dp[i][j] / cnt[i][j];
}
void trans(){
    for(int i = 0; i < 8; i++){
        for(int j = 0; j < 8; j++){
            dp[i][j] = tmp[i][j];
        }
    }
}
signed main(){
    int n;
    double ans = 0;
    cin >> n;
    build();
    for(int x = 0; x < 8; x++){
        for(int y = 0; y < 8; y++){
            reset();
            dp[x][y] = 1;
            for(int k = 0; k < n; k++){
                clear();
                for(int i = 0; i < 8; i++){
```

```
                    for(int j = 0; j < 8; j++){
                        move(i, j);
                    }
                }
                trans();
            }
            for(int i = 0; i < 8; i++){
                for(int j = 0; j < 8; j++){
                    E[i][j] *= (1.0 - dp[i][j]);
                }
            }
        }
    }
    for(int i = 0; i < 8; i++){
        for(int j = 0; j < 8; j++){
            ans += E[i][j];
        }
    }
    cout << fixed << setprecision(6) << ans;
    return 0;
}
```

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

機率

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

signed main(){
    int n;
    double k, ans = 0, u, d;
    cin >> n >> k;
    for(double i = 1; i <= k; i += 1){
        u = d = 1;
        for(int j = 0; j < n; j++){
            u *= i / k;
            d *= (i - 1) / k;
        }
        ans += i * (u - d);
    }
    cout << fixed << setprecision(6) << ans;
    return 0;
}
```

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

機率

```cpp
#include <bits/stdc++.h>
using namespace std;
array<double, 104> R;
signed main(){
    int n;
    double ans = 0;
    cin >> n;
    for(int i = 0; i < n; i++){
        cin >> R[i];
    }
    for(int i = 0; i < n; i++){
        for(int j = i + 1; j < n; j++){
            for(double k = 2; k <= R[i]; k += 1){
                ans += min(k - 1, R[j]) / R[j] / R[i];
            }
        }
    }
    cout << fixed << setprecision(6) << ans;
    return 0;
}
```

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

賽局 DP

```cpp
#include <bits/stdc++.h>
using namespace std;
array<int, 104> P;
array<bool, 1000004> dp;
signed main(){
    int n, k;
    cin >> n >> k;
    for(int i = 0; i < k; i++) cin >> P[i];
    sort(P.begin(), P.begin() + k);
    for(int i = 1; i <= n; i++){
        for(int p : P){
            if(p > i || !p) break;
            dp[i] |= !dp[i - p];
        }
        cout << (dp[i]? "W" : "L");
    }
    return 0;
}
```

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

賽局

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

signed main(){
    int t, n, x, ans;
    cin >> t;
    while(t--){
        cin >> n;
        ans = 0;
        while(n--){
            cin >> x;
            ans ^= x;
        }
        cout << (ans? "first\n" : "second\n");
    }
    return 0;
}
```

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

賽局

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

signed main(){
    int t, n, x, ans;
    cin >> t;
    while(t--){
        cin >> n;
        ans = 0;
        while(n--){
            cin >> x;
            ans ^= (x % 4);
        }
        cout << (ans? "first\n" : "second\n");
    }
    return 0;
}
```

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

賽局

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

signed main(){
    int t, n, p, ans;
    cin >> t;
    while(t--){
        cin >> n;
        ans = 0;
        for(int i = 1; i <= n; i++){
            cin >> p;
            if(~i & 1) ans ^= p;
        }
        cout << (ans? "first\n" : "second\n");
    }
    return 0;
}
```

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

```cpp
#include <bits/stdc++.h>
using namespace std;
array<int, 2004> SG;
int mex(set<int> &S){
    for(int i = 0; i <= 2000; i++){
        if(S.find(i) == S.end()) return i;
    }
}
void build(int n){
    set<int> S;
    for(int i = 1; i <= n; i++){
        S.clear();
        for(int j = 1; j < i; j++){
            if(j != i - j) S.insert(SG[i - j] ^ SG[j]);
        }
        SG[i] = mex(S);
    }
}
signed main(){
    int t, n;
    cin >> t;
    build(2000);
    while(t--){
        cin >> n;
        if(n > 2000) cout << "first\n";
        else cout << (SG[n]? "first\n" : "second\n");
    }
    return 0;
}
```

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

賽局

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

signed main(){
    int t, n, x, ans;
    cin >> t;
    while(t--){
        cin >> n;
        ans = 0;
        for(int i = 0; i < n; i++){
            cin >> x;
            ans |= x & 1;
        }
        cout << (ans? "first\n" : "second\n");
    }
    return 0;
}
```

String Algorithms

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

Trie DP

```cpp
#include <bits/stdc++.h>
#define int long long
using namespace std;
const int mod = 1e9 + 7;
int p = 0;
string S;
array<array<int, 26>, 1000004> trie;
array<int, 1000004> cnt;
array<int, 5004> dp;
void update(string s){
    int u = 0;
    for(int i = 0; i < s.size(); i++){
        if(!trie[u][s[i] - 'a']) trie[u][s[i] - 'a'] = ++p;
        u = trie[u][s[i] - 'a'];
    }
    cnt[u]++;
}
int query(int i){
    int u = 0, ans = 0;
    for(; i < S.size(); i++){
        if(!trie[u][S[i] - 'a']) return ans;
        u = trie[u][S[i] - 'a'];
        ans += (cnt[u] * dp[i + 1]) % mod;
        ans %= mod;
    }
    return ans;
}
signed main(){
    int k;
    string K;
    cin >> S >> k;
    while(k--){
        cin >> K;
        update(K);
    }
    dp[S.size()] = 1;
    for(int i = S.size() - 1; i >= 0; i--){
        dp[i] += query(i);
    }
    cout << dp[0];
    return 0;
}
```

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

KMP

```cpp
#include <bits/stdc++.h>
using namespace std;
array<int, 1000004> F;
void build(string T){
    int p;
    F[0] = -1;
    for(int i = 1; i < T.size(); i++){
        p = F[i - 1];
        while(~p && T[i] != T[p + 1]) p = F[p];
        if(T[i] == T[p + 1]) p++;
        F[i] = p;
    }
}
int match(string T, string S){
    int p = -1, cnt = 0;
    for(int i = 0; i < S.size(); i++){
        while(~p && S[i] != T[p + 1]) p = F[p];
        if(S[i] == T[p + 1]) p++;
        if(p + 1 == T.size()) cnt++, p = F[p];
    }
    return cnt;
}
signed main(){
    string S, T;
    cin >> S >> T;
    build(T);
    cout << match(T, S);
    return 0;
}
```

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

Z

```cpp
#include <bits/stdc++.h>
using namespace std;
array<int, 1000004> Z;
signed main(){
    string S;
    int l = 0, r = 0;
    cin >> S;
    Z[0] = S.size();
    for(int i = 1; i < S.size(); i++){
        if(i + Z[i - l] <= r) Z[i] = Z[i - l];
        else{
            l = i;
            if(i > r) r = i;
            while(r < S.size() && S[r] == S[r - l]) r++;
            r--;
            Z[i] = r - l + 1;
        }
    }
    for(int i = S.size() - 1; i > 0; i--){
        if(Z[i] == .size() - i) cout << Z[i] << " ";
    }
    return 0;
}
```

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

Z

```cpp
#include <bits/stdc++.h>
using namespace std;
array<int, 1000004> Z;
signed main(){
    int l = 0, r = 0;
    string S;
    cin >> S;
    Z[0] = S.size();
    for(int i = 1; i < S.size(); i++){
        if(i + Z[i - l] <= r) Z[i] = Z[i - l];
        else{
            l = i;
            if(i > r) r = i;
            while(r < S.size() && S[r] == S[r - l]) r++;
            r--;
            Z[i] = r - l + 1;
            if(Z[i] == S.size() - i) cout << i << " ";
        }
    }
    cout << S.size();
    return 0;
}
```

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

Booth

```cpp
#include <bits/stdc++.h>
using namespace std;
string LMS(string S){
    int n = S.size(), i = 0, j = 1, k;
    S += S;
    while(i < n && j < n){
        k = 0;
        while(S[i + k] == S[j + k]) k++;
        if(S[i + k] <= S[j + k]) j += k + 1;
        else i += k + 1;
        if(i == j) j++;
    }
    i = i < n? i : j;
    return S.substr(i, n);
}
signed main(){
    string S;
    cin >> S;
    cout << LMS(S);
    return 0;
}
```

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

Manacher

```cpp
#include <bits/stdc++.h>
#define pb push_back
#define mid (l + r) / 2
using namespace std;
array<int, 2000004> P;
string LPS(string T){
    string S, A;
    int l = 0, r = 0, lng = 1, ans = 0;
    for(int i = 0; i <= 2 * T.size(); i++){
        if(i & 1) S += T[i / 2];
        else S += '#';
    }
    P[0] = 1;
    for(int i = 1; i < S.size(); i++){
        P[i] = max(min(r - i, P[2 * mid - i]), 1);
        while(i >= P[i] && i + P[i] < S.size() && S[i - P[i]] == S[
            l = i - P[i];
            r = i + P[i];
            P[i]++;
        }
        if(P[i] > lng || (P[i] == lng && i & 1)){
            lng = P[i];
            ans = i;
        }
    }
    for(int i = ans - lng + 1; i < ans + lng; i++){
        if(i & 1) A += S[i];
    }
    return A;
}
signed main(){
    string S;
    cin >> S;
    cout << LPS(S);
    return 0;
}
```

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

KMP DP

```cpp
#include <bits/stdc++.h>
#define int long long
using namespace std;
const int mod = 1e9 + 7;
array<int, 104> F;
array<array<int, 104>, 1004> dp;
void KMP(string &S){
    int p = 0;
    for(int i = 2; i < S.size(); i++){
        while(p && S[i - 1] != S[p]) p = F[p];
        if(S[i - 1] == S[p]) p++;
        F[i] = p;
    }
}
int DP(string &S, int n, int m){
    int p;
    dp[0][0] = 1;
    for(int i = 1; i <= m; i++){
        for(int j = 0; j < n; j++){
            for(char k = 'A'; k <= 'Z'; k++){
                p = j;
                while(p && S[p] != k) p = F[p];
                if(S[p] == k) p++;
                dp[i][p] += dp[i - 1][j];
                dp[i][p] %= mod;
            }
        }
        dp[i][n] += dp[i - 1][n] * 26;
        dp[i][n] %= mod;
    }
    return dp[m][n];
}
signed main(){
    int n, m;
    string S;
    cin >> m >> S;
    n = S.size();
    KMP(S);
    cout << DP(S, n, m);
    return 0;
}
```

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

Hash 線段樹

```cpp
#include <bits/stdc++.h>
#define int long long
#define mid ((l + r) >> 1)
#define lc (p << 1)
#define rc ((p << 1) | 1)
using namespace std;
const int mod = 1e9 + 7;
array<int, 200004> H;
array<int, 800004> F, B;
void ha(int n){
    H[0] = 1;
    for(int i = 1; i <= n; i++){
        H[i] = (H[i - 1] * 29) % mod;
    }
}
void pull(int p, int l, int r){
    F[p] = (F[lc] + H[mid - l + 1] * F[rc]) % mod;
    B[p] = (B[rc] + H[r - mid] * B[lc]) % mod;
}
void update(int p, int c, int x, int l, int r){
    if(c < l || c > r) return;
    if(l == r){
        F[p] = B[p] = x;
        return;
    }
    update(lc, c, x, l, mid);
    update(rc, c, x, mid + 1, r);
    pull(p, l, r);
}
int query(int p, int ql, int qr, int l, int r, bool t){
    if(ql > r || qr < l) return 0;
    if(t){
        if(ql <= l && qr >= r) return F[p];
        return (query(lc, ql, qr, l, mid, t) + H[max(0ll, mid - max
    }else{
        if(ql <= l && qr >= r) return B[p];
        return (query(rc, ql, qr, mid + 1, r, t) + H[max(0ll, min(r
    }
}
signed main(){
    int n, q, t, l, r, k, f, b;
    char x;
    cin >> n >> q;
    ha(n);
    for(int i = 1; i <= n; i++){
        cin >> x;
        update(1, i, x - 'a', 1, n);
    }
    while(q--){
        cin >> t;
        if(t == 1){
            cin >> k >> x;
            update(1, k, x - 'a', 1, n);
```

```
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

```cpp
#include <bits/stdc++.h>
#define pb push_back
#define mid (l + r) / 2
using namespace std;
array<int, 100004> SA, RNK, F, L;
array<vector<int>, 100004> buk;
void sort(array<int, 100004> &A, int n){
    int cnt = 0;
    for(int i = 0; i < n; i++){
        buk[A[SA[i]]].pb(SA[i]);
    }
    for(int i = 0; i < max(n, 26); i++){
        for(int x : buk[i]){
            SA[cnt++] = x;
        }
        buk[i].clear();
    }
}
void suf(string &S){
    int n = S.size(), ff = -1, ll = -1, cnt = -1;
    for(int i = 0; i < n; i++){
        F[i] = S[i] - 'a';
        SA[i] = i;
    }
    sort(F, n);
    for(int i = 0; i < n; i++){
        if(F[SA[i]] == ff && L[SA[i]] == ll) RNK[SA[i]] = cnt;
        else RNK[SA[i]] = ++cnt;
        ff = F[SA[i]], ll = L[SA[i]];
    }
    for(int j = 1; j < n; j <<= 1){
        cnt = ff = ll = -1;
        for(int i = 0; i < n; i++){
            F[i] = RNK[i];
            L[i] = i + j < n? RNK[i + j] : 0;
        }
        sort(L, n);
        sort(F, n);
        for(int i = 0; i < n; i++){
            if(F[SA[i]] == ff && L[SA[i]] == ll) RNK[SA[i]] = cnt;
            else RNK[SA[i]] = ++cnt;
            ff = F[SA[i]], ll = L[SA[i]];
        }
    }
}
bool cmp(string &T, string &S, int k){
    for(int i = 0; i < T.size() && i + k < S.size(); i++){
        if(T[i] < S[k + i]) return 1;
        else if(T[i] > S[k + i]) return 0;
    }
    if(T.size() > S.size() - k) return 0;
    return 1;
}
```

```
bool BS(string &S, string &T){
    int l = 0, r = S.size() - 1;
    while(l != r){
        if(cmp(T, S, SA[mid])) r = mid;
        else l = mid + 1;
    }
    if(T.size() > S.size() - SA[l]) return 0;
    for(int i = 0; i < T.size(); i++){
        if(T[i] != S[SA[l] + i]) return 0;
    }
    return 1;
}
signed main(){
    int k;
    string S, T;
    cin >> S >> k;
    suf(S);
    while(k--){
        cin >> T;
        cout << (BS(S, T)? "YES\n" : "NO\n");
    }
    return 0;
}
```

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

Suffix Array

```cpp
#include <bits/stdc++.h>
#define pb push_back
#define mid (l + r) / 2
using namespace std;
array<int, 100004> SA, RNK, F, L;
array<vector<int>, 100004> buk;
void sort(array<int, 100004> &A, int n){
    int cnt = 0;
    for(int i = 0; i < n; i++){
        buk[A[SA[i]]].pb(SA[i]);
    }
    for(int i = 0; i < max(n, 26); i++){
        for(int x : buk[i]){
            SA[cnt++] = x;
        }
        buk[i].clear();
    }
}
void suf(string &S){
    int n = S.size(), cnt = -1, ff = -1, ll = -1;
    for(int i = 0; i < n; i++){
        SA[i] = i;
        F[i] = S[i] - 'a';
    }
    sort(F, n);
    for(int i = 0; i < n; i++){
        if(F[SA[i]] == ff && L[SA[i]] == ll) RNK[SA[i]] = cnt;
        else RNK[SA[i]] = ++cnt;
        ff = F[SA[i]], ll = L[SA[i]];
    }
    for(int j = 1; j < n; j <<= 1){
        cnt = ff = ll = -1;
        for(int i = 0; i < n; i++){
            F[i] = RNK[i];
            L[i] = i + j < n? RNK[i + j] : 0;
        }
        sort(L, n);
        sort(F, n);
        for(int i = 0; i < n; i++){
            if(F[SA[i]] == ff && L[SA[i]] == ll) RNK[SA[i]] = cnt;
            else RNK[SA[i]] = ++cnt;
            ff = F[SA[i]], ll = L[SA[i]];
        }
    }
}
bool cmp(string &S, string &T, int k, bool t){
    for(int i = 0; i < T.size() && i + k < S.size(); i++){
        if(T[i] < S[i + k]) return 1;
        else if(T[i] > S[i + k]) return 0;
    }
    if(T.size() > S.size() - k) return 0;
    if(t) return 0;
    else return 1;
}
```

```
}
int BS(string &S, string &T, bool t){
    int l = 0, r = S.size() - 1;
    while(l != r){
        if(cmp(S, T, SA[mid], t)) r = mid;
        else l = mid + 1;
    }
    return l;
}
signed main(){
    int k;
    string S, T;
    cin >> S >> k;
    S += '~';
    suf(S);
    while(k--){
        cin >> T;
        cout << BS(S, T, 1) - BS(S, T, 0) << "\n";
    }
    return 0;
}
```

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

```cpp
#include <bits/stdc++.h>
#define pb push_back
#define mid ((l + r) >> 1)
#define lc (p << 1)
#define rc ((p << 1) | 1)
using namespace std;
array<int, 400004> seg;
array<int, 100004> SA, RNK, F, L;
array<vector<int>, 100004> buk;
void update(int p, int c, int x, int l, int r){
    if(c < l || c > r) return;
    if(l == r){
        seg[p] = x;
        return;
    }
    update(lc, c, x, l, mid);
    update(rc, c, x, mid + 1, r);
    seg[p] = min(seg[lc], seg[rc]);
}
int query(int p, int ql, int qr, int l, int r){
    if(ql > r || qr < l) return 1e9;
    if(ql <= l && qr >= r) return seg[p];
    return min(query(lc, ql, qr, l, mid), query(rc, ql, qr, mid + 1
}
void sort(array<int, 100004> &A, int n){
    int cnt = 0;
    for(int i = 0; i < n; i++){
        buk[A[SA[i]]].pb(SA[i]);
    }
    for(int i = 0; i < max(n, 26); i++){
        for(int x : buk[i]){
            SA[cnt++] = x;
        }
        buk[i].clear();
    }
}
void suf(string &S){
    int n = S.size(), cnt = -1, ff = -1, ll = -1;
    for(int i = 0; i < n; i++){
        SA[i] = i;
        F[i] = S[i] - 'a';
    }
    sort(F, n);
    for(int i = 0; i < n; i++){
        if(F[SA[i]] == ff && L[SA[i]] == ll) RNK[SA[i]] = cnt;
        else RNK[SA[i]] = ++cnt;
        ff = F[SA[i]], ll = L[SA[i]];
    }
    for(int j = 1; j < n; j <<= 1){
        cnt = ll = ff = -1;
        for(int i = 0; i < n; i++){
            F[i] = RNK[i];
            L[i] = i + j < n? RNK[i + j] : 0;
```

```
        }
        sort(L, n);
        sort(F, n);
        for(int i = 0; i < n; i++){
            if(F[SA[i]] == ff && L[SA[i]] == ll) RNK[SA[i]] = cnt;
            else RNK[SA[i]] = ++cnt;
            ff = F[SA[i]], ll = L[SA[i]];
        }
    }
    for(int i = 0; i < n; i++){
        update(1, i, SA[i], 0, S.size() - 1);
    }
}
bool cmp(string &S, string &T, int k, bool t){
    for(int i = 0; i < T.size() && i + k < S.size(); i++){
        if(T[i] < S[i + k]) return 1;
        else if(T[i] > S[i + k]) return 0;
    }
    if(T.size() > S.size() - k) return 0;
    return t ^ 1;
}
int BS(string &S, string &T, bool t){
    int l = 0, r = S.size() - 1;
    while(l != r){
        if(cmp(S, T, SA[mid], t)) r = mid;
        else l = mid + 1;
    }
    return l;
}
int pos(string &S, string &T){
    if(BS(S, T, 1) == BS(S, T, 0)) return -1;
    return query(1, BS(S, T, 0), BS(S, T, 1) - 1, 0, S.size() - 1)
}
signed main(){
    int k;
    string S, T;
    cin >> S >> k;
    S += '~';
    for(int &s : seg) s = 1e9;
    suf(S);
    while(k--){
        cin >> T;
        cout << pos(S, T) << "\n";
    }
    return 0;
}
```

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

Suffix Array

```
#include <bits/stdc++.h>
#define pb push_back
#define int long long
using namespace std;
array<int, 100004> SA, RNK, F, L, LCP;
array<vector<int>, 100004> buk;
void sort(array<int, 100004> &A, int n){
    int cnt = 0;
    for(int i = 0; i < n; i++){
        buk[A[SA[i]]].pb(SA[i]);
    }
    for(int i = 0; i < max(n, 26ll); i++){
        for(int x : buk[i]){
            SA[cnt++] = x;
        }
        buk[i].clear();
    }
}
void suf(string &S){
    int n = S.size(), cnt = -1, ff = -1, ll = -1;
    for(int i = 0; i < n; i++){
        SA[i] = n - i - 1;
        F[i] = S[i] - 'a';
    }
    sort(F, n);
    for(int i = 0; i < n; i++){
        if(F[SA[i]] == ff && L[SA[i]] == ll) RNK[SA[i]] = cnt;
        else RNK[SA[i]] = ++cnt;
        ff = F[SA[i]], ll = L[SA[i]];
    }
    for(int j = 1; j < n; j <<= 1){
        cnt = ff = ll = -1;
        for(int i = 0; i < n; i++){
            F[i] = RNK[i];
            L[i] = i + j < n? RNK[i + j] : 0;
        }
        sort(L, n);
        sort(F, n);
        for(int i = 0; i < n; i++){
            if(F[SA[i]] == ff && L[SA[i]] == ll) RNK[SA[i]] = cnt;
            else RNK[SA[i]] = ++cnt;
            ff = F[SA[i]], ll = L[SA[i]];
        }
    }
}
int cp(string &S){
    int n = S.size(), lcp = 0, k, sum = 0;
    for(int i = 0; i < n; i++){
        RNK[SA[i]] = i;
    }
    for(int i = 0; i < n; i++){
        if(!RNK[i]) continue;
        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

```cpp
#include <bits/stdc++.h>
#define pb push_back
using namespace std;
array<int, 100004> SA, RNK, F, L;
array<vector<int>, 100004> buk;
void sort(array<int, 100004> &A, int n){
    int cnt = 0;
    for(int i = 0; i < n; i++){
        buk[A[SA[i]]].pb(SA[i]);
    }
    for(int i = 0; i < max(n, 26); i++){
        for(int x : buk[i]){
            SA[cnt++] = x;
        }
        buk[i].clear();
    }
}
void suf(string &S){
    int n = S.size(), cnt = -1, ff = -1, ll = -1;
    for(int i = 0; i < n; i++){
        SA[i] = n - i - 1;
        F[i] = S[i] - 'a';
    }
    sort(F, n);
    for(int i = 0; i < n; i++){
        if(F[SA[i]] == ff && L[SA[i]] == ll) RNK[SA[i]] = cnt;
        else RNK[SA[i]] = ++cnt;
        ff = F[SA[i]], ll = L[SA[i]];
    }
    for(int j = 1; j < n; j <<= 1){
        cnt = ll = ff = -1;
        for(int i = 0; i < n; i++){
            F[i] = RNK[i];
            L[i] = i + j < n? RNK[i + j] : 0;
        }
        sort(L, n);
        sort(F, n);
        for(int i = 0; i < n; i++){
            if(F[SA[i]] == ff && L[SA[i]] == ll) RNK[SA[i]] = cnt;
            else RNK[SA[i]] = ++cnt;
            ff = F[SA[i]], ll = L[SA[i]];
        }
    }
}
string lcp(string &S){
    int n = S.size(), cp = 0, k, lng = 0, ans;
    for(int i = 0; i < n; i++){
        RNK[SA[i]] = i;
    }
    for(int i = 0; i < n; i++){
        if(!RNK[i]) continue;
        k = SA[RNK[i] - 1];
        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

```cpp
#include <bits/stdc++.h>
using namespace std;
array<int, 1000004> Z, F;
void ZZZ(string &S){
    int l = 0, r = 0;
    Z[0] = 0;
    for(int i = 1; i < S.size(); i++){
        if(i + Z[i - l] < r) Z[i] = Z[i - l];
        else{
            l = i;
            if(i > r) r = i;
            while(r < S.size() && S[r] == S[r - l]) r++;
            Z[i] = r - l;
        }
    }
}
void KMP(string &S){
    int p;
    F[0] = -1;
    for(int i = 1; i < S.size(); i++){
        p = F[i - 1];
        while(~p && S[p + 1] != S[i]) p = F[p];
        if(S[i] == S[p + 1]) p++;
        F[i] = p;
    }
}
signed main(){
    string S;
    cin >> S;
    ZZZ(S);
    for(int i = 0; i < S.size(); i++){
        cout << Z[i] << " ";
    }
    cout << "\n";
    KMP(S);
    for(int i = 0; i < S.size(); i++){
        cout << F[i] + 1 << " ";
    }
    return 0;
}
```

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

Suffix Array

```cpp
#include <bits/stdc++.h>
#define int long long
#define pb push_back
#define mid (l + r) / 2
using namespace std;
array<int, 100004> SA, RNK, LCP, F, L;
array<vector<int>, 100004> buk;
void sort(array<int, 100004> &A, int n){
    int cnt = 0;
    for(int i = 0; i < n; i++){
        buk[A[SA[i]]].pb(SA[i]);
    }
    for(int i = 0; i < max(n, 26ll); i++){
        for(int x : buk[i]){
            SA[cnt++] = x;
        }
        buk[i].clear();
    }
}
void suf(string &S){
    int n = S.size(), cnt = -1, ff = -1, ll = -1;
    for(int i = 0; i < n; i++){
        SA[i] = n - i - 1;
        F[i] = S[i] - 'a';
    }
    sort(F, n);
    for(int i = 0; i < n; i++){
        if(F[SA[i]] == ff && L[SA[i]] == ll) RNK[SA[i]] = cnt;
        else RNK[SA[i]] = ++cnt;
        ff = F[SA[i]], ll = L[SA[i]];
    }
    for(int j = 1; j < n; j <<= 1){
        cnt = ff = ll = -1;
        for(int i = 0; i < n; i++){
            F[i] = RNK[i];
            L[i] = i + j < n? RNK[i + j] : 0;
        }
        sort(L, n);
        sort(F, n);
        for(int i = 0; i < n; i++){
            if(F[SA[i]] == ff && L[SA[i]] == ll) RNK[SA[i]] = cnt;
            else RNK[SA[i]] = ++cnt;
            ff = F[SA[i]], ll = L[SA[i]];
        }
    }
}
void lcp(string &S){
    int n = S.size(), cp = 0, k;
    for(int i = 0; i < n; i++){
        RNK[SA[i]] = i;
    }
    for(int i = 0; i < n; i++){
        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)

```cpp
#include <bits/stdc++.h>
#define int long long
#define pb push_back
#define mid ((l + r) >> 1)
#define lc (p << 1)
#define rc ((p << 1) + 1)
using namespace std;
array<int, 100004> SA, RNK, F, L;
array<int, 400004> seg, tag;
array<vector<int>, 100004> buk;
void pull(int p){
    seg[p] = seg[lc] + seg[rc];
}
void push(int p, int l, int r){
    seg[lc] += (mid - l + 1) * tag[p];
    seg[rc] += (r - mid) * tag[p];
    tag[lc] += tag[p];
    tag[rc] += tag[p];
    tag[p] = 0;
}
void update(int p, int ql, int qr, int v, int l, int r){
    if(ql > r || qr < l) return;
    if(ql <= l && qr >= r){
        seg[p] += (r - l + 1) * v;
        tag[p] += v;
        return;
    }
    push(p, l, r);
    update(lc, ql, qr, v, l, mid);
    update(rc, ql, qr, v, mid + 1, r);
    pull(p);
}
int query(int p, int ql, int qr, int l, int r){
    if(ql > r || qr < l) return 0;
    if(ql <= l && qr >= r) return seg[p];
    push(p, l, r);
    return query(lc, ql, qr, l, mid) + query(rc, ql, qr, mid + 1,
}
void sort(array<int, 100004> &A, int n){
    int cnt = 0;
    for(int i = 0; i < n; i++){
        buk[A[SA[i]]].pb(SA[i]);
    }
    for(int i = 0; i < max(n, 26ll); i++){
        for(int x : buk[i]) SA[cnt++] = x;
        buk[i].clear();
    }
}
void suf(string &S){
    int n = S.size(), cnt, ff, ll;
    for(int i = 0; i < n; i++){
       SA[i] = n - i - 1;
       RNK[i] = F[i] = S[i] - 'a';
```

```
    }
    sort(F, n);
    for(int k = 1; k < n; k <<= 1){
        cnt = ff = ll = -1;
        for(int i = 0; i < n; i++){
            F[i] = RNK[i];
            L[i] = i + k < n? RNK[i + k] : 0;
        }
        sort(L, n);
        sort(F, n);
        for(int i = 0; i < n; i++){
            if(F[SA[i]] == ff && L[SA[i]] == ll) RNK[SA[i]] = cnt;
            else RNK[SA[i]] = ++cnt;
            ff = F[SA[i]], ll = L[SA[i]];
        }
    }
    for(int i = 0; i < n; i++){
        update(1, i, i, n - SA[i], 0, n);
    }
}
int bis(string &S, int t, int l, int r, char c){
    int p = l - 1;
    for(int i = 1 << 16; i > 0; i >>= 1){
        if(p + i <= r && S[SA[p + i] + t] <= c) p += i;
    }
    return p;
}
char cha(string &S, int k, int t, int l, int r){
    int p, n = S.size();
    char cl = 'a', cr = 'z', cm;
    while(cl != cr){
        cm = (cl + cr) >> 1;
        p = bis(S, t, l, r, cm);
        if(query(1, l, p, 0, n) < k) cl = cm + 1;
        else cr = cm;
    }
    return cl;
}
string see(string &S, int k){
    int n = S.size(), l = 0, r = n - 1, t = 0, tmp;
    char p;
    string s;
    while(k > 0){
        p = cha(S, k, t, l, r);
        tmp = l;
        l = bis(S, t, l, r, p - 1) + 1;
        k -= query(1, tmp, l - 1, 0, n);
        r = bis(S, t, l, r, p);
        update(1, l, r, -1, 0, n);
        k -= r - l + 1;
        s += p;
        t++;
    }
```

```
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

```cpp
#include <bits/stdc++.h>
#define int long long
#define pb push_back
using namespace std;
array<int, 100004> SA, RNK, F, L, sum;
array<vector<int>, 100004> buk;
void sort(array<int, 100004> &A, int n){
    int cnt = 0;
    for(int i = 0; i < n; i++){
        buk[A[SA[i]]].pb(SA[i]);
    }
    for(int i = 0; i < max(n, 26ll); i++){
        for(int x : buk[i]){
            SA[cnt++] = x;
        }
        buk[i].clear();
    }
}
void suf(string &S){
    int n = S.size(), cnt, ff, ll;
    for(int i = 0; i < n; i++){
        SA[i] = n - i - 1;
        RNK[i] = F[i] = S[i] - 'a';
    }
    sort(F, n);
    for(int j = 1; j < n; j <<= 1){
        cnt = ff = ll = -1;
        for(int i = 0; i < n; i++){
            F[i] = RNK[i];
            L[i] = i + j < n? RNK[i + j] : 0;
        }
        sort(L, n);
        sort(F, n);
        for(int i = 0; i < n; i++){
            if(F[SA[i]] == ff && L[SA[i]] == ll) RNK[SA[i]] = cnt;
            else RNK[SA[i]] = ++cnt;
            ff = F[SA[i]], ll = L[SA[i]];
        }
    }
}
void lcp(string &S){
    int n = S.size(), cp = 0, k;
    for(int i = 0; i < n; i++) RNK[SA[i]] = i;
    for(int i = 0; i < n; i++){
        if(!RNK[i]){
            sum[0]++;
            sum[n - i]--;
            continue;
        }
        k = SA[RNK[i] - 1];
        if(cp) cp--;
        while(S[i + cp] == S[k + cp]) cp++;
        sum[cp]++;
```

```
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

```cpp
#include <bits/stdc++.h>
#define int long long
using namespace std;
struct vec{
    int x, y;
    vec(int x, int y): x(x), y(y){}
    vec operator+(vec v){
        return vec(x + v.x, y + v.y);
    }
    vec operator-(vec v){
        return vec(x - v.x, y - v.y);
    }
    int operator*(vec v){
        return x * v.x + y * v.y;
    }
    int operator^(vec v){
        return x * v.y - y * v.x;
    }
};
signed main(){
    int t, x1, y1, x2, y2, x3, y3, ans;
    cin >> t;
    while(t--){
        cin >> x1 >> y1 >> x2 >> y2 >> x3 >> y3;
        ans = vec(x2 - x1, y2 - y1) ^ vec(x3 - x1, y3 - y1);
        if(ans > 0) cout << "LEFT\n";
        else if(ans < 0) cout << "RIGHT\n";
        else cout << "TOUCH\n";
    }
    return 0;
}
```

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

向量

```cpp
#include <bits/stdc++.h>
#define int long long
using namespace std;
struct vec{
    int x, y;
    vec(){}
    vec(int x, int y): x(x), y(y){}
    vec operator+(vec v){
        return vec(x + v.x, y + v.y);
    }
    vec operator-(vec v){
        return vec(x - v.x, y - v.y);
    }
    int operator*(vec v){
        return x * v.x + y * v.y;
    }
    int operator^(vec v){
        return x * v.y - y * v.x;
    }
};
int cal(vec a, vec b, vec c, bool t){
    int ans;
    if(t) ans = (c - a) ^ (b - a);
    else ans = (c - a) * (b - a);
    if(ans == 0) return 0;
    else if(ans > 0) return 1;
    else return -1;
}
bool line(vec a, vec b, vec c, vec d){
    bool ans = 1;
    ans &= !cal(a, b, c, 1) && !cal(a, b, d, 1);
    ans &= (a - c) * (b - d) > 0 && (a - d) * (b - c) > 0;
    return ans;
}
bool inter(vec a, vec b, vec c, vec d){
    bool ans = 1;
    ans &= cal(a, b, c, 1) * cal(a, b, d, 1) <= 0;
    ans &= cal(c, d, a, 1) * cal(c, d, b, 1) <= 0;
    ans &= !line(a, b, c, d);
    return ans;
}
signed main(){
    int t, x1, y1, x2, y2, x3, y3, x4, y4;
    cin >> t;
    while(t--){
        cin >> x1 >> y1 >> x2 >> y2 >> x3 >> y3 >> x4 >> y4;
        cout << (inter(vec(x1, y1), vec(x2, y2), vec(x3, y3), vec(x
    }
    return 0;
}
```

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

向量　行列式

```cpp
#include <bits/stdc++.h>
#define int long long
using namespace std;
struct vec{
    int x, y;
    vec(){}
    vec(int x, int y): x(x), y(y){}
    vec operator+(vec v){
        return vec(x + v.x, y + v.y);
    }
    vec operator-(vec v){
        return vec(x - v.x, y - v.y);
    }
    int operator*(vec v){
        return x * v.x + y * v.y;
    }
    int operator^(vec v){
        return x * v.y - y * v.x;
    }
};
array<vec, 1004> E;
int area(int n){
    int ans = 0;
    for(int i = 0; i < n; i++){
        ans += E[i] ^ E[i + 1];
    }
    return abs(ans);
}
signed main(){
    int n, x, y;
    cin >> n;
    for(int i = 0; i < n; i++){
        cin >> x >> y;
        E[i] = vec(x, y);
    }
    E[n] = E[0];
    cout << area(n);
    return 0;
}
```

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

向量

```cpp
#include <bits/stdc++.h>
#define int long long
using namespace std;
struct vec{
    int x, y;
    vec(){}
    vec(int x, int y): x(x), y(y){}
    vec operator+(vec v){
        return vec(x + v.x, y + v.y);
    }
    vec operator-(vec v){
        return vec(x - v.x, y - v.y);
    }
    int operator*(vec v){
        return x * v.x + y * v.y;
    }
    int operator^(vec v){
        return x * v.y - y * v.x;
    }
};
const vec inf = vec(1, 4e9);
array<vec, 1004> V;
int cal(vec a, vec b, vec c){
    int ans;
    ans = (c - a) ^ (b - a);
    if(ans > 0) return 1;
    else if(ans < 0) return -1;
}
int inter(vec a, vec b, vec c, vec d){
    int abc, abd, cda, cdb;
    abc = cal(a, b, c);
    abd = cal(a, b, d);
    cda = cal(c, d, a);
    cdb = cal(c, d, b);
    if(abc * abd < 0 && cda * cdb < 0) return 1;
    else return 0;
}
bool bound(vec a, vec b, vec c){
    if((a - b) ^ (c - b)) return 0;
    if((a - b) * (c - b) > (c - b) * (c - b)) return 0;
    if((a - b) * (c - b) < 0) return 0;
    return 1;
}
int in(vec p, int n){
    int cnt = 0;
    bool on = 0;
    vec end = p + inf;
    for(int i = 0; i < n; i++){
        cnt += inter(p, end, V[i], V[i + 1]);
        on |= bound(p, V[i], V[i + 1]);
    }
    if(on) return 0;
    if(cnt & 1) return 1;
```

```
    else return -1;
}
signed main(){
    int n, m, x, y, ans;
    cin >> n >> m;
    for(int i = 0; i < n; i++){
        cin >> x >> y;
        V[i] = vec(x, y);
    }
    V[n] = V[0];
    while(m--){
        cin >> x >> y;
        ans = in(vec(x, y), n);
        if(ans > 0) cout << "INSIDE\n";
        else if(ans < 0) cout << "OUTSIDE\n";
        else cout << "BOUNDARY\n";
    }
    return 0;
}
```

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

向量　皮克定理

```cpp
#include <bits/stdc++.h>
#define int long long
using namespace std;
struct vec{
    int x, y;
    vec(){}
    vec(int x, int y): x(x), y(y){}
    vec operator+(vec v){
        return vec(x + v.x, y + v.y);
    }
    vec operator-(vec v){
        return vec(x - v.x, y - v.y);
    }
    int operator*(vec v){
        return x * v.x + y * v.y;
    }
    int operator^(vec v){
        return x * v.y - y * v.x;
    }
};
array<vec, 100004> V;
int gcd(int a, int b){
    return b? gcd(b, a % b) : a;
}
void point(int n){
    int area = 0, on = 0, in;
    vec tmp;
    for(int i = 0; i < n; i++){
        tmp = V[i] - V[i + 1];
        area += V[i] ^ V[i + 1];
        on += gcd(abs(tmp.x), abs(tmp.y));
    }
    area = abs(area);
    in = (area - on + 2) / 2;
    cout << in << " " << on;
}
signed main(){
    int n, x, y;
    cin >> n;
    for(int i = 0; i < n; i++){
        cin >> x >> y;
        V[i] = vec(x, y);
    }
    V[n] = V[0];
    point(n);
    return 0;
}
```

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

掃描線

```cpp
#include <bits/stdc++.h>
#define int long long
#define vec pair<int, int>
#define x first
#define y second
using namespace std;
array<vec, 200004> V;
int sq(int x){
    return ceil(sqrt(x));
}
int dis(vec a, vec b){
    return (a.x - b.x) * (a.x - b.x) + (a.y - b.y) * (a.y - b.y);
}
int mindis(int n){
    int p = 0, d = 8e18;
    set<vec> S;
    for(int i = 0; i < n; i++){
        while(V[p].x <= V[i].x - d){
            S.erase({V[p].y, V[p].x});
            p++;
        }
        for(auto it = S.upper_bound({V[i].y - sq(d), V[i].x}); it->
            if(it == S.end()) break;
            d = min(d, dis(*it, {V[i].y, V[i].x}));
        }
        S.insert({V[i].y, V[i].x});
    }
    return d;
}
signed main(){
    int n, x, y;
    cin >> n;
    for(int i = 0; i < n; i++){
        cin >> x >> y;
        V[i] = {x, y};
    }
    sort(V.begin(), V.begin() + n);
    cout << mindis(n);
    return 0;
}
```

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

向量 Monoton Stack

```cpp
#include <bits/stdc++.h>
#define int long long
#define pb push_back
#define ppb pop_back
using namespace std;
struct vec{
    int x, y;
    vec(){}
    vec(int x, int y): x(x), y(y){}
    vec operator+(vec v){
        return vec(x + v.x, y + v.y);
    }
    vec operator-(vec v){
        return vec(x - v.x, y - v.y);
    }
    int operator*(vec v){
        return x * v.x + y * v.y;
    }
    int operator^(vec v){
        return x * v.y - y * v.x;
    }
};
array<vec, 200004> V;
vector<vec> S;
bool cmp(vec a, vec b){
    return a.x == b.x? a.y < b.y : a.x < b.x;
}
void print(vec v){
    cout << v.x << " " << v.y << "\n";
}
bool comp(vec a, vec b){
    if(a ^ b) return (a ^ b) > 0;
    return (a * b) > 0;
}
void hull(int n, int s){
    vec a, b;
    for(int i = 0; i < n; i++){
        while(S.size() > s){
            b = S.back();
            S.ppb();
            a = S.back();
            if(comp(V[i] - a, b - a)){
                S.pb(b);
                break;
            }
        }
        S.pb(V[i]);
    }
    S.ppb();
}
signed main(){
    int n, x, y;
    cin >> n;
```

```
    for(int i = 0; i < n; i++){
        cin >> x >> y;
        V[i] = vec(x, y);
    }
    sort(V.begin(), V.begin() + n, cmp);
    hull(n, 1);
    reverse(V.begin(), V.begin() + n);
    hull(n, S.size() + 1);
    cout << S.size() << "\n";
    for(vec s : S) print(s);
    return 0;
}
```