

1

□□	□□
int	$[-2^{31}, 2^{31}]$
long long	$[-2^{63}, 2^{63} - 1]$

1

□□□□	□□
Bit□□□□	□□□□□□□□□□□□□□□□
Byte□□□□	□□□□□□□□1 Byte = 8 Bits

1


```
##include<iostream>
using namespace std;
const int N = 1e6 + 10;

int n;
int q[N];

void quick_sort(int q[],int l,int r){
    if(l >= r) return;
    int x = q[(int)((l + r) / 2)];
    int i = l - 1,j = r + 1;
    while(i < j){
        do i++; while(q[i] < x);
        do j--; while(q[j] > x);
        if(i < j) swap(q[i],q[j]);
    }
    quick_sort(q,l,j);
    quick_sort(q,j + 1,r);
}
```

```

int main(){
    scanf("%d", &n);
    for(int i = 0;i < n;i++) scanf("%d",&q[i]);

    quick_sort(q,0,n - 1);

    for(int i = 0;i < n;i++) printf("%d ",q[i]);

    return 0;
}

```

퀵정렬

- 시간복잡도 \$O(n \log n)\$
- 1. 중앙값 찾기 \$mid = \frac{l + r}{2}\$
- 2. 분할 정복
- 3. 재귀 호출
 - 원소 1개인 경우
 - 원소 2개인 경우
 - 원소 3개인 경우

```

##include<iostream>
using namespace std;

const int N = 1e6 + 10;
int n;
int q[N],temp[N];

void merge_sort(int q[],int l,int r){
    if(l >= r) return;
    int mid = (l + r) / 2;
    merge_sort(q, l, mid);
    merge_sort(q, mid + 1, r);

    int k = 0,i = l,j = mid + 1;
    while(i <= mid && j <= r){
        if(q[i] <= q[j]) temp[k++] = q[i++];
        if(q[i] > q[j]) temp[k++] = q[j++];
    }
    while(i <= mid) temp[k++] = q[i++];
    while(j <= r) temp[k++] = q[j++];

    for(int i = l, j = 0;i <= r;i++,j++) q[i] = temp[j];
}

int main(){
    scanf("%d",&n);

    for(int i = 0;i < n;i++) scanf("%d",&q[i]);
}

```

```

    merge_sort(q,0,n - 1);

    for(int i = 0;i < n;i++) printf("%d ",q[i]);

    return 0;
}

```

二分查找

```

// 检查是否满足条件
bool check(int x){...}

// 二分查找check1
int binary_search1(int l,int r){
    while(l < r){
        int mid = (l + r) / 2;
        if(check(mid)) r = mid;
        else l = mid + 1;
    }
    return l;
}

// 二分查找check2
int binary_search2(int l, int r){
    while(l < r){
        int mid = (l + r + 1) / 2; // 防止溢出
        if(check(mid)) l = mid;
        else r = mid - 1;
    }
    return l;
}

```

二分查找

```

bool check(double x){}

// 二分查找
double binary_search(double l, double r){
    while(r - l > eps){ // eps精度控制
        double mid = (l + r) / 2;
        if(check(mid)) r = mid;
        else l = mid;
    }
    return l;
}

```

二分

二、字符串相加

- 例题

```
##include<iostream>
##include<vector>
const int N = 1e6 + 10;
int n;
using namespace std;

vector<int> add(vector<int> &a, vector<int> &b) { //字符串相加
    vector<int> c;
    int t = 0; //进位
    for(int i = 0; i < a.size() || i < b.size(); i++) {
        if(i < a.size()) t += a[i];
        if(i < b.size()) t += b[i];
        c.push_back(t % 10);
        t /= 10;
    }
    if(t) c.push_back(1);
    return c
}
int main(){
    string a, b;
    vector<int> A, B;
    cin >> a >> b;
    for(int i = a.size() - 1; i >= 0; i--) A.push_back(a[i] - '0');
    for(int i = b.size() - 1; i >= 0; i--) B.push_back(b[i] - '0');
    vector<int> c = add(A, B);
    for(int i = c.size() - 1; i >= 0; i--) printf("%d", c[i]);
}
```

- 例题

- \$A > B\$ ⇔ \$A-B\$
- \$A < B\$ ⇔ \$-(B-A)\$

```
##include<iostream>
##include<vector>
const int N = 1e6 + 10;
int n;
using namespace std;

// 字符串比较
bool cmp(vector<int> &a, vector<int> &b){
    if(a.size() != b.size()) return a.size() > b.size();
    for(int i = a.size() - 1; i >= 0; i++) {
        if(a[i] != b[i]) return a[i] > b[i];
    }
    return true;
}
```

```

vector<int> sub(vector<int> &a, vector<int> &b){
    vector<int> c;
    int t = 0;
    for(int i = 0; i < a.size(); i++){
        t = a[i] - t;
        if(i < b.size()) t -= b[i];
        c.push_back((t + 10) % 10);
        if(t < 0) t = 1;
        else t = 0;
    }
    while(c.size() > 1 && c.back() == 0) c.pop_back();
    return c;
}

int main(){
    String a, b;
    vector<int> A, B;
    cin >> a >> b;
    for(int i = a.size() - 1; i >= 0; i--) A.push_back(a[i] - '0');
    for(int i = b.size() - 1; i >= 0; i--) B.push_back(b[i] - '0');
    if(cmp(A,B)){
        vector<int> c = sub(A,B);
        for(int i = c.size() - 1; i >= 0; i--) printf("%d", c[i]);
    }
    else{
        vector<int> c = sub(B,A);
        printf("-");
        for(int i = c.size() - 1; i >= 0; i--) printf("%d", c[i]);
    }
}

```

- 乘法

```

vector<int> multi(vector<int> &a, int b){
    vector<int> c;
    int t = 0;
    for(int i = 0; i < a.size(); i++){
        t += a[i] * b;
        c.push_back(t % 10);
        t = t / 10;
    }
    while(t > 0){
        c.push_back(t % 10);
        t /= 10;
    }
    // if(t > 0) c.push_back(t);
    while(c.size() > 1 && c.back() == 0) c.pop_back();
    return c;
}

```

-

`vector<int> div(vector<int> &a, int b, int &r){`
 `vector<int> c;`
 `r = 0;`
 `for(int i = a.size() - 1; i >= 0; i--){`
 `r = r * 10 + a[i];`
 `c.push_back(r / b);`
 `r %= b;`
 `}`
 `while(t > 0){`
 `c.push_back(t)`
 `}`
 `while(c.size() > 1 && c.back() == 0) c.pop_back();`
 `reverse(c.begin(), c.end());`

}

•

-

`a_1, a_2, \dots, a_n`

`S_i = a_1 + a_2 + a_3 + \dots + a_i`

`S_k - S_l`

```
int t;
int b[n];
// s0 = 0 n+1
void sum(int a[]){
    int t = 0;
    for(int i = 0 ; i <= n; i++){
        b[i] = t;
        t += a[i];
    }
}
```

-

- $S_{ij} = \sum_{m \leq i, n \leq j} a_{mn}$
- $a[1][1]$
- $S_{ij} - S_{in} - S_{mj} + S_{mn}$
- $S_{ij} = a_{ij} + S_{i-1,j} + S_{i,j-1} - S_{i-1,j-1}$
- $S_{0,1} - S_{0,0} - S_{1,0}$

```

int s[M][N];
int main(){
    s[0][0] = 0;
    for(int i = 1;i < n;i++){
        for(int j = 1;j < m;j++){
            scanf("%d",&a[i][j]);
            s[i][j] = s[i][j - 1] + a[i][j] - s[i - 1][j - 1] + s[i - 1][j];
        }
    }
}

```

•

-

a_1, a_2, \dots, a_n b_1, b_2, \dots, b_n $a_i = b_1 + b_2 + \dots + b_i$ b_j a_{n-1} $a_1 = a_1$ $b_2 = a_2 - a_1$ $b_3 = a_3 - a_2$ \dots $b_n = a_n - a_{n-1}$

$$S[i,r] = a_i + C S[r+1, n] O(N) = a_i + b_{r+1} + C S[r+1, n]$$

$$C S[r+1, n] = C S[r+1, r+1] - C S[r+1, r]$$

-

a_{ij}

b_{ij}

```

##include<iostream>
using namespace std;

int n,m,q;

const int N = 1010;
int a[N][N],b[N][N];

void insert(int x1,int y1,int x2,int y2,int c){
    b[x1][y1] += c;
    b[x2 + 1][y2 + 1] += c;
    b[x1][y2 + 1] -= c;
    b[x2 + 1][y1] -= c;
}

int main(){
    cin >> n >> m >> q;
    for(int i = 1;i <= n;i++){
        for(int j = 1;j <= m;j++){
            int t;
            cin >> t;
            insert(i,j,i,j,t);
        }
    }
    while(q--){
        int x1,y1,x2,y2,c;
        cin >> x1 >> y1 >> x2 >> y2 >> c;
        insert(x1,y1,x2,y2,c);
    }
}

```

```

int x1,y1,x2,y2,c;
cin >> x1 >> y1 >> x2 >> y2 >> c;
insert(x1,y1,x2,y2,c);
}

for(int i = 1;i <= n;i++){
    for(int j = 1;j <= m;j++){
        b[i][j] = b[i][j] + b[i - 1][j] + b[i][j - 1] - b[i - 1][j - 1]; // 累加
    }
}

for(int i = 1;i <= n;i++){
    for(int j = 1;j <= m;j++){
        cout << b[i][j] << " ";
    }
    cout << endl;
}
}

```

累加

$O(n^2)$

时间复杂度

常数

$n \log n$ k 常数

常数

1. $k \gg n >> k$
 2. $n \& 1$
 3. $n >> k \& 1;$
- $\text{lowbit}(x) = x \& (-x)$
 - $x = 1010 \dots 10$
 - $x = 101000 \dots 1000$
 - $x \& -x$

常数

常数时间复杂度

常数时间复杂度

- all
- add
- query

常数

- all
- add
- query

- 亂數生成器

```

##include<iostream>
##include<vector>
##include<algorithm>
using namespace std;

const int N = 3e5 + 10; // 亂數生成器
int a[N], s[N]; // 亂數生成器

vector<int> alls; // 亂數生成器

typedef pair<int,int> PII;
vector<PII> add,query;
int n,m;
// 亂數生成器
int find(int x){
    int l = 0, r = alls.size() - 1;
    while(l < r){
        int mid = l + r >> 1;
        if(alls[mid] >= x) r = mid;
        else l = mid + 1;
    }
    return r + 1;
}

int main(){
    cin >> n >> m;
    // 亂數
    for(int i = 0;i < n;i++){
        int x, c;
        cin >> x >> c;
        alls.push_back(x);
        add.push_back({x,c});
    }

    // 亂數
    while(m--){
        int l,r;
        cin >> l >> r;
        query.push_back({l,r});
        alls.push_back(l);
        alls.push_back(r);
    }

    // 亂數
    sort(alls.begin(),alls.end());
    alls.erase(unique(alls.begin(),alls.end()),alls.end());
}

// 亂數

```

```

for(auto x: add){
    int t = find(x.first);
    a[t] += x.second;
}

// 亂數
for(int i = 1;i <= alls.size();i++){
    s[i] = a[i] + s[i - 1];
}

// 結果
for(auto x: query){
    int l = find(x.first);
    int r = find(x.second);
    cout << s[r] - s[l - 1] << endl;
}
}

```

問題

問題文

1. 問題文
2. 問題文

```

##include<iostream>
##include<algorithm>
##include<vector>
using namespace std;
const int N = 100010;

vector<pair<int,int>> t;

int n;

void merge(vector<pair<int,int>> &s){
    vector<pair<int,int>> temp;
    sort(s.begin(),s.end());
    // 亂數
    int l = s[0].first,r = s[0].second;
    for(int i = 1;i < s.size();i++){
        if(s[i].first > r) {
            temp.push_back({l,r});
            l = s[i].first;
            r = s[i].second;
        }
        else r = max(r,s[i].second); // 結果
    }
    // 亂數
    temp.push_back({l,r});
}

```

```

        t = temp;
    }

int main(){
    cin >> n;
    while(n--){
        int a,b;
        cin >> a >> b;
        t.push_back({a,b});
    }
    merge(t);
    cout << t.size();
}

```

□□□□

□□□□□□□

□□

1. □□□□□□□□□□□□□□
 - e[N]□□□□□□□□
 - ne[N]□□□□□□□□next□□
2. □□□□□□□□□□□□□□
 - l[N]□□□□□□□□
 - r[N]□□□□□□□□
3. □□□□n□□□□

□□□

```

##include<iostream>

using namespace std;

const int N = 1000010;
/*
e[N]□□□□□□□□
ne[N]□□□□□□□□
head□□□□□
idx□□□□□□□□□□
*/
int e[N],ne[N];
int head,idx;

void init(){
    head = -1;
    idx = 0;
}

```

```

void add_to_head(int c){
    e[idx] = c;
    ne[idx] = head;
    head = idx++;
}

void erase(int k){
    // 从k指向的结点开始断开
    if(k == -1) {
        head = ne[head];
    }
    ne[k] = ne[ne[k]];
}

void add(int x,int c){
    e[idx] = c;
    ne[idx] = ne[x];
    ne[x] = idx++;
}

int main(){
    init();
    int m;
    cin >> m;
    while(m--){
        int k,x;
        char op;
        cin >> op;
        if(op == 'H'){
            cin >> x;
            add_to_head(x);
        }
        else if(op == 'D'){
            cin >> k ;
            erase(k - 1);
        }
        else if(op == 'I'){
            cin >> k >> x;
            add(k - 1,x);
        }
    }
    for(int i = head;i != -1;i = ne[i]){
        cout << e[i] << " ";
    }
}

```

```

##include<iostream>
using namespace std;

const int N = 100010;
int l[N],r[N],e[N];
int idx;
// 00000000000000001
void init(){
    r[0] = 1,l[1] = 0;
    idx = 2;
}
// 00000k00000
void add(int k,int x){
    e[idx] = x;
    r[idx] = r[k];
    l[idx] = k;
    l[r[k]] = idx;
    r[k] = idx;
    idx++;
}
void remove(int k){
    l[r[k]] = l[k];
    r[l[k]] = r[k];
}

int main(){
    int m;
    init();
    cin >> m;
    while(m--){
        string op;
        cin >> op;
        int k,x;
        if(op == "L"){
            cin >> x;
            add(0,x);
        }
        else if(op == "R"){
            cin >> x;
            add(l[1],x);
        }
        else if(op == "D"){
            cin >> k;
            remove(k + 1); // 01000000000idx02
        }
        else if(op == "IL"){
            cin >> k >> x;
            add(l[k + 1],x);
        }else if(op == "IR"){

```

```

        cin >> k >> x;
        add(k + 1,x);
    }
}
for(int i = r[0];i != 1;i = r[i]){
    cout << e[i] << " ";
}
}

```

□

□□□□

```

##include<iostream>
using namespace std;

const int N = 100010;
int s[N];
int top;

void init(){
    top = -1;
}

bool isempty(){
    return top == -1;
}

void push(int x){
    s[++top] = x;
}

void pop(){
    top--;
}

int query(){
    return s[top];
}

int main(){
    init();
    int m;
    cin >> m;
    while(m--){
        int x;
        string op;
        cin >> op;
        if(op == "push"){
            cin >> x;
            push(x);
        }
    }
}

```

```

    }
    else if(op == "pop") pop();
    else if(op == "empty"){
        if(isempty()) cout << "YES" << endl;
        else cout << "NO" << endl;
    }
    else if(op == "query") cout << query()<< endl;
}
}

```

□□

□□□□

```

##include<iostream>
using namespace std;

const int N = 100010;
int s[N],top,tail;

void init(){
    top = 0;
    tail = -1;
}

void push(int x){
    s[++tail] = x;
}

bool isempty(){
    return tail < top;
}

void pop(){
    top++;
}

int query(){
    return s[top];
}

int main(){
    int m;
    init();
    cin >> m;
    while(m--){
        int x;
        string op;
        cin >> op;

```

```
if(op == "push"){
    cin >> x;
    push(x);
}
else if(op == "pop"){
    pop();
}
else if(op == "empty"){
    if(isempty()) cout << "YES" << endl;
    else cout << "NO" << endl;
}
else if( op == "query"){
    cout << query() << endl;
}
}
```

10

 <https://www.acwing.com/video/5400/>

```
#include<iostream>

using namespace std;
const int N = 1e5 + 10;
int st[N];
int top = -1;
int n;
// 二分查找
int main(){
    cin >> n;
    for(int i = 0;i < n;i++){
        int x;
        cin >> x;
        while(top >= 0 && st[top] >= x) top--;
        if(top != -1) cout << st[top] << " ";
        else cout << -1 << " ";
        st[++top] = x;
    }
}
```

□□□□□□□□□□□□

- 

```
##include<iostream>
using namespace std;

const int N = 1e6 +
```

```

int a[N],st[N];
int head = 0,tail = -1;
int main(){

    int n,k;
    cin >> n >> k;
    for(int i = 0;i < n;i++){
        cin >> a[i];
    }
    // 1111
    for(int i = 0;i < n;i++){
        if( i - k + 1 > st[head]) head++;
        while(tail >= head && a[st[tail]] >= a[i]) tail--;
        st[++tail] = i;
        if(i >= k - 1) cout << a[st[head]] << " ";
    }
    cout << endl;
    // 1111
    head = 0,tail = -1;
    for(int i = 0;i < n;i++){
        if(i - k + 1 > st[head]) head++;
        while(tail >= head && a[st[tail]] <= a[i]) tail--;
        st[++tail] = i;
        if(i >= k - 1) cout << a[st[head]] << " ";
    }
    cout << endl;
}

```

1111111111

```

##include<iostream>

using namespace std;
const int N = 1e6 + 10;
int n,k;
int tail = -1,head = 0;
int q[N],a[N];

int main(){
    cin >> n >> k;
    for(int i = 1;i <= n;i++){
        cin >> a[i];
    }
    for(int i = 1;i <= n;i++){
        if(head <= tail && i > k && q[head] == a[i - k]) head++;
        while(tail >= head && q[tail] > a[i]) tail--;
        q[++tail] = a[i];
        if(i > k - 1){
            cout << q[head] << " ";
        }
    }
}

```

```

cout << endl;
head = 0, tail = -1;
for(int i = 1;i <= n;i++){
    if(head <= tail && i > k && q[head] == a[i - k]) head++;
    while(tail >= head && q[tail] < a[i]) tail--;
    q[++tail] = a[i];
    if(i > k - 1){
        cout << q[head] << " ";
    }
}
}

```

KMP

- 亂數生成器
- 亂數生成器next
- 亂數n個亂數生成器 n + 1 亂數生成器亂數生成器 n - next[n] 亂數生成器 n+1 亂數

```

##include<iostream>
using namespace std;

const int N = 1e6 + 10;
char s[N],p[N];
// 亂數生成器1亂數生成器0亂數生成器1亂數生成器0ne[1]乱数
int ne[N];
int main(){
    int n,m;
    cin >> n >> p + 1 >> m >> s + 1;
    /*
    j乱数生成器i乱数生成器乱数生成器乱数
        a b a b a
        1 2 3 4 5
    ne  0 0 1 2 3
        a b a b a
        a b a b a
    */
    for(int i = 2,j = 0;i <= n;i++){
        while(j && p[i] != p[j + 1]) j = ne[j];
        if(p[i] == p[j + 1]) j++;
        ne[i] = j;
    }

    for(int i = 1,j = 0; i <= m;i++){
        while(j && s[i] != p[j + 1]) j = ne[j];
        if(s[i] == p[j + 1]) j++;
        if(j == n) {
            // 亂數生成器1亂數生成器0亂數生成器i-n+1
            cout << i - n << " ";
            j = ne[j];
        }
    }
}

```

```
    }
}

}
```

Trie

字符串字典树

字符串



字符串字典树
字符串字典树

字符串字典树**0**

```
##include<iostream>
##include<string.h>
using namespace std;

const int N = 1e5 + 10;
int son[N][26],idx,cnt[N];
int n;

void insert(char str[]){
    int p = 0;
    for(int i = 0;str[i];i++){
        if(!son[p][str[i] - 'a']) son[p][str[i] - 'a'] = ++idx;
        p = son[p][str[i] - 'a'];
    }
    cnt[p]++;
}

int query(char str[]){
    int p = 0;
    for(int i = 0;str[i];i++){
        if(!son[p][str[i] - 'a']) return 0;
        p = son[p][str[i] - 'a'];
    }
    return cnt[p];
}

int main(){
    ios::sync_with_stdio(false);
    cout.tie(0),cin.tie(0);
    cin >> n;
    while(n--){
        string s;
        char x[N];
```

```

    cin >> s;
    if(s == "I"){
        cin >> x;
        insert(x);
    }
    else if(s == "Q"){
        cin >> x;
        cout << query(x) << endl;
    }
}
}

```

}

████████

```

##include<iostream>
using namespace std;

const int N = 1e5 + 10;
const int M = 31 * N; //████████████████████31*N████
int n;
int a[N],son[M][2];
int idx = 0;
void insert(int x){
    int p = 0;
    for(int i = 30;i >= 0;i--){
        int u = x >> i & 1;
        if(!son[p][u]) son[p][u] = ++idx;
        p = son[p][u];
    }
}

int search(int x){
    int p = 0,res = 0;
    for(int i = 30;i >= 0;i--){
        int u = x >> i & 1;
        if(son[p][!u]){
            p = son[p][!u]; // █████████████████████100001
            res = res * 2 + 1; //0100 -> 1001 █████ (100)_2 * 2 + 1 = (1001)_2 04 *
2 + 1 = 9
        }
        else {
            p = son[p][u];// █████████████████████████████████
            res = res * 2;
        }
    }
    return res;
}

int main(){

```

```

scanf("%d",&n);
int ans = 0;
for(int i = 0;i < n;i++) {
    scanf("%d",&a[i]);
    insert(a[i]);
}
for(int i = 0;i < n;i++) {
    ans = max(ans,search(a[i]));
}
cout << ans;

```

問題

1. 木構造
2. データ構造

問題

- 木構造
- データ構造
- $p[x] \times p[y] \rightarrow y \in p[x] = y$
- データ構造

問題

- データ構造のsize

問題

```

##include<iostream>
using namespace std;

const int N = 1e5 + 10;
int n,m;

int h[N];

int find(int x){
    if(x != h[x]) h[x] = find(h[x]);
    return h[x];
}

int main(){
    ios::sync_with_stdio(false);
    cin.tie(0),cout.tie(0);
    cin >> n >> m;
    for(int i = 0;i < n;i++) h[i] = i; // 初期化
    int a,b;
    while(m--){
        char c;
        cin >> c;
        if(c == 'M'){

```

```
    cin >> a >> b;
    if(find(a) == find(b)) continue;
    h[find(a)] = find(b);
}
else{
    cin >> a >> b;
    if(find(a) == find(b)) cout << "Yes" << endl;
    else cout << "No" << endl;
}
}
```

1

1

1. □□□□□
 2. □□□□□□□
 3. □□□□□
 4. □□□□□□□□□□□□

2

- $x \square \square \square \square 2x$
 - $x \square \square \square \square 2x+1$

7

- $\Omega n / 2^{\Omega(n)}$
 - $\Omega(n^2) \text{ up } \Omega(n^{\Omega(\log \log n)})$
 - $\Omega(n^2) \text{ down } \Omega(n^{\Omega(\log \log n)})$

1

1

1. 亂數產生
 - 亂數產生範例
 - 亂數產生範例
 - 亂數find範例
 - 亂數×亂數範例×亂數範例
 - 亂數×亂數範例×亂數範例

```
// 二进制  
##include<iostream>
```

```
##include<cstring>
using namespace std;

const int N = 2e5 + 3; // 0x3f3f3f3f+30000000000000000000
const int NuLL = 0x3f3f3f3f; //0000
int h[N];
int n;

// 00000000x000000000000x00000000
int find(int x){
    int k = (x % N + N) % N;
    while(h[k] != NuLL && h[k] != x){
        k++;
        if(k == N) k = 0;
    }
    return k;
}

int main(){
    memset(h,0x3f,sizeof(h)); // 000
    string s;
    cin >> n;
    while(n--){
        cin >> s;
        int x;
        if(s == "I"){
            cin >> x;
            h[find(x)] = x;
        }
        else if(s == "Q"){
            cin >> x;
            int k = find(x);
            if(h[k] == x) cout << "Yes" << endl;
            else cout << "No" << endl;
        }
    }
}
```

2. □□□

- 一二三四五六七八九
 - 一二三四五六七八九
 - 一二三四五六七八九

```
// 用于读取文件  
##include<iostream>  
##include<cstring>  
using namespace std;
```

```

const int N = 1e5 + 3; // 100000
int h[N];
int e[N], ne[N];
int idx = 0;
int n;

void insert(int x){
    int k = (x % N + N) % N; // 0到N-1的索引，x在[-N, N]范围内
    e[idx] = x;
    ne[idx] = h[k];
    h[k] = idx++;
}

bool find(int x){
    int k = (x % N + N) % N;
    for(int i = h[k]; i != -1; i = ne[i]){
        if(e[i] == x) return true;
    }
    return false;
}

int main(){
    memset(h, -1, sizeof(h)); // 初始化所有节点的前驱为-1
    cin >> n;
    string s;
    int x;
    while(n--){
        cin >> s;
        if(s == "I"){
            cin >> x;
            insert(x);
        }
        else if(s == "Q"){
            cin >> x;
            if(find(x)) cout << "Yes" << endl;
            else cout << "No" << endl;
        }
    }
}
}

```

•

•

- p13113331
- p13113331
- q\$2^{64}\$
- unsigned long long
- h

- $h[0] = 0$
 - $h[i] = h[i - 1] * 131 + str[i];$
 - 亂数生成の複雑度はO(1)である
 - $h[L-R] = h[R] - h[L-1] * 131^{\{R-L+1\}}$
 - $131^{\{R-L+1\}}$ の計算
 - $h[R]$ と $h[L-1]$ の差を求めるには $h[L-1]$ の値を記憶する

```
##include<iostream>
using namespace std;

const int N = 1e5 + 10;
char s[N];
unsigned long long P[N],h[N];
int n,m;
int p = 131;

int search(int l,int r){
    return h[r] - h[l - 1] * P[r - l + 1];
}

int main(){
    cin >> n >> m;
    cin >> s;
    P[0] = 1;
    for(int i = 0;i < n;i++){
        P[i + 1] = P[i] * p;
        h[i + 1] = h[i] * p + s[i];
    }
}

while(m --){
    int l1,r1,l2,r2;
    cin >> l1 >> r1 >> l2 >> r2;
    if(search(l1,r1) == search(l2,r2)) cout << "Yes" << endl;
    else cout << "No" << endl;
}
```

STL

vector

- `vector`
 - `vector<int> a = (10,3) vector[10][3]`
 - `size()`
 - `empty()`
 - `clear()`
 - `front() / back()`
 - `push_back() / pop_back()`
 - `begin() / end()`

pair<int,int>

- `pair::pair()`
- `first()/second()` 1/2
- `p = (20,"zxb")` `p = make_pair(20,"zxb")`

string

- `substr()`
 - `substr(1,2)` 1 2
- `size()`
- `empty()`
- `clear()`

queue

- `size()`
- `empty()`
- `push()`
- `front()`
- `back()`
- `pop()`

priority_queue

•

- `push()`
- `front()`
- `pop()`

deque

- `size()`
- `empty()`
- `clear()`
- `front()`
- `back()`
- `push_back() /pop_back()`
- `push_front() /pop_front()`
- `begin() /end()`

stack

- `size()`
- `empty()`
- `push()`
- `top()`
- `pop()`

set

専用メソッド

- `size()` サイズ取得
- `empty()` 空か否か
- `clear()` 全削除
- `insert()` 要素挿入
- `find()` 要素検索
- `count()` 要素数取得
- `erase()`
 - 要素削除(x)
 - 要素削除範囲
- `lower_bound()` 要素検索x範囲開始位置end()
- `upper_bound()` 要素検索x範囲終了位置end()
- `begin()/end()` 0番目/範囲終了位置

map

- `size()` サイズ取得
- `empty()` 空か否か
- `clear()` 全削除
- `insert()` 要素挿入pair
- `erase()` 要素削除pair
- `m['key']` 要素keyのvalue
- `lower_bound()` 要素検索x範囲開始位置end()
- `upper_bound()` 要素検索x範囲終了位置end()

multiset

専用メソッド

- `size()` サイズ取得
- `empty()` 空か否か
- `clear()` 全削除
- `insert()` 要素挿入
- `find()` 要素検索
- `count()` 要素数取得
- `erase()`
 - 要素削除(x)
 - 要素削除範囲
- `lower_bound()` 要素検索x範囲開始位置end()
- `upper_bound()` 要素検索x範囲終了位置end()

multimap

- `size()` サイズ取得
- `empty()` 空か否か
- `clear()` 全削除
- `insert()` 要素挿入pair
- `erase()` 要素削除pair
- `m['key']` 要素keyのvalue
- `lower_bound()` 要素検索x範囲開始位置end()
- `upper_bound()` 要素検索x範囲終了位置end()

unordered_map**unordered_set****unordered_multiset****unordered_multimap**

时间复杂度O(1)

lower_bound() upper_bound()

二叉树

DFS

深度优先搜索

遍历所有节点+回溯+剪枝

递归

```
##include<iostream>
using namespace std;
const int N = 10;
int n,t[N];
bool state[N];

void dfs(int p){
    if(p == n){
        for(int i = 0;i < n;i++) cout << t[i] << " ";
        cout << endl;
    }
    for(int i = 1;i <= n;i++){
        if(!state[i]){
            t[p] = i;
            state[i] = true;
            dfs(p + 1);
            state[i] = false;
        }
    }
}

int main(){
    cin >> n;
    dfs(0);
}
```

n个

```
##include<iostream>
using namespace std;

const int N = 12,M = 2 * N;
char d[N][N];
int n;
```

```

bool col[N],e[M],ne[M];
// \u002f
void dfs(int u){
    if(u == n + 1){
        for(int i = 1;i <= n;i++){
            for(int j = 1;j <= n;j++)
                cout << d[i][j];
        cout << endl;
    }
    cout << endl;
    return ;
}

for(int i = 1;i <= n;i++){
    if(!col[i] && !e[i + u] && !ne[n - i + u]){
        col[i] = true,e[i + u] = true,ne[n - i + u] = true;
        d[u][i] = 'Q';
        dfs(u + 1);
        col[i] = false,e[i + u] = false,ne[n - i + u] = false;
        d[u][i] = '.';
    }
}
}

int main(){
    cin >> n;
    for(int i = 1;i <= n;i++){
        for(int j = 1;j <= n;j++)
            d[i][j] = '.';
    }
    dfs(1);
    return 0;
}

```

BFS

1 BFS

三

```
##include<iostream>
using namespace std;
typedef pair<int,int> PII;
const int N = 110;
int d[N][N],p[N][N]; //d[][] p[()][]
PII q[N * N];
bool st[N][N];
int head = 0,tail = -1;
int dx[4] = {0,1,0,-1},dy[4] = {-1,0,1,0};
```

```

int n,m;
int bfs(){
    st[0][0] = true;
    p[0][0] = 0;
    q[++tail] = {0,0};
    while(head <= tail){
        auto x = q[head++];
        for(int i = 0;i < 4;i++){
            int a = x.first + dx[i],b = x.second + dy[i]; // a[][]b[]
            if(a >= 0 && a < n && b >= 0 && b < m && !st[a][b] && d[a][b] == 0){
                st[a][b] = true;
                p[a][b] = p[x.first][x.second] + 1;
                q[++tail] = {a,b};
            }
        }
    }
    return p[n - 1][m - 1];
}
int main(){
    cin >> n >> m;
    for(int i = 0;i < n;i ++){
        for(int j = 0;j < m;j++)
            cin >> d[i][j];
    }
    cout << bfs();
}

```

□□□□□□

```

##include<iostream>
##include<queue>
##include<cstring>
using namespace std;
const int N = 1e5 + 10;
int dist[N],e[N],ne[N],h[N],idx;
bool st[N];
int n,m;
queue<int> q;

void add(int a,int b){
    e[idx] = b,ne[idx] = h[a],h[a] = idx++;
}

int main(){
    cin >> n >> m;
    memset(h,-1,sizeof h);
    memset(dist,0x3f,sizeof(dist));
    while(m--){
        int a,b;
        cin >> a >> b;

```

```

        add(a,b);
    }
    q.push(1);
    dist[1] = 0;
    st[1] = true;
    while(q.size()){
        int t = q.front();
        q.pop();
        for(int i = h[t];i != -1;i = ne[i]){
            int j = e[i];
            if(!st[j]){
                dist[j] = dist[t] + 1;
                q.push(j);
                st[j] = true;
            }
        }
    }
    if(dist[n] == 0x3f3f3f3f) cout << -1;
    else cout << dist[n];
}

```

██████

██ (u→v)████████ u ████████ v ████

████████████

```

##include<iostream>
##include<cstring>
using namespace std;
const int N = 1e5 + 10;
int n,m;
int h[N],e[N],ne[N],idx;
int q[N],d[N];

void add(int a,int b){
    e[idx] = b,ne[idx] = h[a],h[a] = idx++;
}

bool topsort(){
    int head = 0,tail = -1;
    for(int i = 1;i <= n;i++){
        if(!d[i]) q[++tail] = i;
    }
    while(head <= tail){
        int x = q[head++];
        for(int i = h[x];i != -1;i = ne[i]){
            int j = e[i];
            d[j]--;
            if(!d[j]) q[++tail] = j;
        }
    }
}

```

```

        return tail == n - 1;
    }

int main(){
    memset(h,-1,sizeof h);
    cin >> n >> m;
    int a,b;
    for(int i = 1;i <= m;i++){
        cin >> a >> b;
        add(a,b);
        d[b]++;
    }
    if(topsort()){
        for(int i = 0;i < n;i++) cout << q[i] << " ";
    }
    else {
        cout << -1;
    }
}

```

□□□



- Dijkstra**□□□□□□□
- Dijkstra**□□□□□□□□□□□□
- □□□□□□□
- □□□□□□□□□
- □□□□□□□□□
- □□□□□□□□□SPFA

□□**Dijkstra**□□

- □□□□□□□□□□□□
- □□□□□□□□□□□

```

##include<iostream>
##include<cstring>
##include<algorithm>
using namespace std;

const int N = 510,M = 1e5 + 10;
int h[N],e[M],w[M],ne[M],idx;
int n,m;
int dist[N];
bool st[N];

void add(int a,int b,int c){
    e[idx] = b,ne[idx] = h[a],w[idx] = c,h[a] = idx++;
}

```

```

int dijkstra(){
    memset(dist,0x3f,sizeof(dist));
    dist[1] = 0;
    for(int i = 1;i < n;i++){
        int t = -1;
        for(int j = 1;j <= n;j++){
            if(!st[j] && (t == -1 || dist[t] > dist[j]) )
                t = j;
        }
        if (dist[t] == 0x3f3f3f3f) break;
        st[t] = true;
        for(int j = h[t]; j != -1;j = ne[j]){ // j\idx
            int k = e[j]; // k\idx
            dist[k] = min(dist[k],dist[t] + w[j]);
        }
    }
    return dist[n];
}

int main(){
    cin >> n >> m;
    memset(h,-1,sizeof h);
    int a,b,c;
    while(m--){
        cin >> a >> b >> c;
        add(a,b,c);
    }
    int res = dijkstra();
    if(res == 0x3f3f3f3f) cout << -1;
    else cout << res;
}

```

□□□□

- □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
- □□□□□□□□□□□□

```

##include<iostream>
##include<cstring>
##include<algorithm>
##include<queue>
using namespace std;

const int N = 2e5 + 10,M = 2e5 + 10;
int h[N],e[M],w[M],ne[M],idx;
int n,m;
int dist[N];
bool st[N];

typedef pair<int,int> PII;

```

```

void add(int a,int b,int c){
    e[idx] = b,ne[idx] = h[a],w[idx] = c,h[a] = idx++;
}

int dijkstra(){
    memset(dist,0x3f,sizeof(dist));
    dist[1] = 0;
    priority_queue <PII,vector<PII>,greater<PII>> heap;
    heap.push({0,1});

    while(heap.size()){
        auto x = heap.top();
        heap.pop();
        if(st[x.second]) continue;
        st[x.second] = true;
        for(int i = h[x.second];i != -1;i = ne[i]){
            int j = e[i];
            if(dist[j] > x.first + w[i]){
                dist[j] = x.first + w[i];
                heap.push({dist[j],j});
            }
        }
    }

    return dist[n];
}

int main(){
    cin >> n >> m;
    memset(h,-1,sizeof h);
    int a,b,c;
    while(m--){
        cin >> a >> b >> c;
        add(a,b,c);
    }
    int res = dijkstra();
    if(res == 0x3f3f3f3f) cout << -1;
    else cout << res;
}

```

Bellman-ford

- ဗိုလ်ချုပ်အကြောင်းအရာ
 - အမြန်လိုအပ်ခဲ့သူ
- အမြန်လိုအပ်ခဲ့သူ

SPFA

- အမြန်လိုအပ်
- အမြန်လိုအပ်ခဲ့သူ
- အမြန်လိုအပ်ခဲ့သူst

```

##include<iostream>
##include<queue>
##include<cstring>
using namespace std;
const int N = 1e5 + 10;
typedef pair<int,int> PII;
int n,m;

int e[N],ne[N],idx,w[N],h[N],dist[N];
bool st[N];
void add(int x,int y,int z){
    e[idx] = y,ne[idx] = h[x],w[idx] = z,h[x] = idx++;
}

int spfa(){
    memset(dist,0x3f,sizeof dist);
    queue<PII> q;
    dist[1] = 0;
    q.push({0,1}); // 起点0到1
    st[1] = true;
    while(q.size()){
        auto t = q.front();
        q.pop();
        int dis = t.first,vec = t.second;
        st[vec] = false;
        for(int i = h[vec];i != -1;i = ne[i]){
            int j = e[i];
            if(dist[j] > dist[vec] + w[i]){ // 节点dis更新dist[vec]
                dist[j] = dist[vec] + w[i];
                if(!st[j]){
                    q.push({dist[j],j});
                    st[j] = true;
                }
            }
        }
    }
    return dist[n] >= 0x3f3f3f3f / 2; // 判定
}

int main(){
    memset(h,-1,sizeof h);
    cin >> n >> m;
    while(m--){
        int x,y,z;
        cin >> x >> y >> z;
        add(x,y,z);
    }
    if(spfa()) cout << "impossible" << endl;
    else cout << dist[n] << endl;
}

```

- 亂數

 - 亂數生成器cnt
 - 亂數生成器
 - dist

```

##include<iostream>
##include<queue>
##include<cstring>
using namespace std;
const int N = 2010, M = 10010;

int n,m;

int e[M],ne[M],idx,w[M],h[N],dist[N],cnt[N];
bool st[N];
void add(int x,int y,int z){
    e[idx] = y,ne[idx] = h[x],w[idx] = z,h[x] = idx++;
}

bool spfa(){
    queue<int> q;
    for(int i = 1;i <= n;i++){
        q.push(i);
        st[i] = true;
    } // 亂數0~1
    while(q.size()){
        auto t = q.front();
        q.pop();
        st[t] = false;
        for(int i = h[t];i != -1;i = ne[i]){
            int j = e[i];
            if(dist[j] > dist[t] + w[i]){ // 亂數dis[vec]dist[vec]
                dist[j] = dist[t] + w[i];
                cnt[j] = cnt[t] + 1;
                if(cnt[j] >= n) return true;
                if(!st[j]){
                    q.push(j);
                    st[j] = true;
                }
            }
        }
    }
    return false;
}

int main(){
    memset(h,-1,sizeof h);

```

```

    cin >> n >> m;
    while(m--){
        int x,y,z;
        cin >> x >> y >> z;
        add(x,y,z);
    }
    if(spfa()) cout << "Yes" << endl;
    else cout << "No" << endl;
}

```

Floyd

- Floyd
- Floyd-Warshall

```

##include<iostream>
##include<cstring>
##include<algorithm>
using namespace std;

const int N = 210;
int n,m,k;
int d[N][N];

void floyd(){
    for(int k = 1;k <= n;k ++){
        for(int i = 1;i <= n;i++){
            for(int j = 1;j <= n;j++)
                d[i][j] = min(d[i][j],d[i][k] + d[k][j]);
        }
    }
}

int main(){
    memset(d,0x3f,sizeof d);
    cin >> n >> m >> k;
    for(int i = 1;i <= n;i++) d[i][i] = 0;
    while(m--){
        int x,y,z;
        cin >> x >> y >> z;
        d[x][y] = min(d[x][y],z);
    }
    floyd();
    while(k--){
        int x,y;
        cin >> x >> y;
        if(d[x][y] >= 0x3f3f3f3f / 2) cout << "impossible" << endl;
        else cout << d[x][y] << endl;
    }
}

```

```
    }  
}
```