

算法模板及常用

蓝桥知识点

一,常用stl

1,string

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

int main() {
    string s = "hello";

    s += " world!";          // 拼接字符串, s 变成 "hello world!"

    // 截取子串
    string sub = s.substr(0, 5); // 截取 "hello"
    cout << "Substr (0, 5): " << sub << endl;

    // 查找子串位置
    size_t pos = s.find("wo"); // 查找 "wo" 在 s 中的位置
    if (pos != string::npos) {
        cout << "Found 'wo' at position: " << pos << endl; // 输出找到的位置
    } else {
        cout << "'wo' not found!" << endl;
    }

    // 获取字符串的长度
    cout << "Length of string: " << s.length() << endl; // 输出字符串的长度

    return 0;
}
s.pop_back(); // 删除最后一个字符
s.erase(1, 1); // 从位置1开始删除1个字符
erase(起始位置, 删除长度)
char first = s.front(); // 取第一个字符
s.erase(0,1); // 删除第一个字符
int k = stoi(s); // 转换
```

2,vector

```

#include<iostream>
#include<string>
#include <vector>
#include <algorithm> // 包含排序所需的头文件
using namespace std;

int main() {
    vector<int> v;           // 创建一个空的 vector
    v.push_back(1);         // 尾部插入元素 1
    v.push_back(3);         // 尾部插入元素 3
    v.push_back(2);         // 尾部插入元素 2

    v.pop_back();           // 尾部删除一个元素（删除 2）

    cout << "Size of vector: " << v.size() << endl; // 输出当前 vector 的大小

    v[0] = 5;               // 修改 vector 第一个元素为 5

    sort(v.begin(), v.end()); // 排序 vector 中的元素

    cout << "Sorted vector: ";
    for (int num : v) {
        cout << num << " "; // 输出排序后的 vector 元素
    }
    cout << endl;

    return 0;
}

```

3,stack

```

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

int main() {
    // 创建一个整数栈
    stack<int> st;

    // 入栈操作
    st.push(10); // 将 10 入栈
    st.push(20); // 将 20 入栈
    st.push(30); // 将 30 入栈

    // 查看栈顶元素

```

```

cout << "Top element: " << st.top() << endl; // 输出栈顶元素 30

// 出栈操作
st.pop(); // 弹出栈顶元素 30
cout << "After pop, top element: " << st.top() << endl; // 输出新的栈顶元素
20

// 再次查看栈顶元素
cout << "Top element after another pop: " << st.top() << endl; // 输出新的
栈顶元素 10

// 你也可以查看栈是否为空
if (st.empty()) {
    cout << "The stack is empty." << endl;
} else {
    cout << "The stack is not empty." << endl;
}

return 0;
}

```

4.queue

```

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

int main() {
    // 创建一个整数队列
    queue<int> q;

    // 入队操作
    q.push(10); // 将 10 入队
    q.push(20); // 将 20 入队
    q.push(30); // 将 30 入队

    // 查看队首元素
    cout << "Front element: " << q.front() << endl; // 输出队首元素 10

    // 出队操作
    q.pop(); // 弹出队首元素 10
    cout << "After pop, front element: " << q.front() << endl; // 输出新的队首
元素 20

    // 再次查看队首元素
    q.pop(); // 弹出队首元素 20
}

```

```

    cout << "After another pop, front element: " << q.front() << endl; // 输出
    新的队首元素 30

    // 检查队列是否为空
    if (q.empty()) {
        cout << "The queue is empty." << endl;
    } else {
        cout << "The queue is not empty." << endl;
    }

    return 0;
}

```

5.priority_queue

```

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

int main() {
    // 默认是大根堆
    priority_queue<int> max_pq;

    // 插入元素到大根堆
    max_pq.push(3);
    max_pq.push(1);
    max_pq.push(2);

    cout << "大根堆操作: " << endl;
    cout << "堆顶元素 (最大元素): " << max_pq.top() << endl; // 输出 3
    max_pq.pop(); // 移除最大元素 3
    cout << "移除最大元素后, 堆顶元素: " << max_pq.top() << endl; // 输出 2

    // 创建小根堆
    priority_queue<int, vector<int>, greater<int>> min_pq;

    // 插入元素到小根堆
    min_pq.push(3);
    min_pq.push(1);
    min_pq.push(2);

    cout << "\n小根堆操作: " << endl;
    cout << "堆顶元素 (最小元素): " << min_pq.top() << endl; // 输出 1
    min_pq.pop(); // 移除最小元素 1
    cout << "移除最小元素后, 堆顶元素: " << min_pq.top() << endl; // 输出 2
}

```

```
    return 0;
}
```

6,set

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

int main() {
    set<int> s;

    // 插入元素
    s.insert(5); // 插入元素 5
    s.insert(2); // 插入元素 2
    s.insert(8); // 插入元素 8
    s.insert(3); // 插入元素 3
    s.insert(1); // 插入元素 1

    // 打印集合中的元素
    cout << "集合中的元素: " << endl;
    for (auto it = s.begin(); it != s.end(); ++it) {
        cout << *it << " ";
    }
    cout << endl;

    // 删除元素 5
    s.erase(5);
    cout << "删除 5 后的集合: " << endl;
    for (auto it = s.begin(); it != s.end(); ++it) {
        cout << *it << " ";
    }
    cout << endl;

    // 检查元素 5 是否存在
    if (s.count(5) > 0) {
        cout << "元素 5 存在" << endl;
    } else {
        cout << "元素 5 不存在" << endl; // 这行将被打印
    }

    // 查找第一个大于等于 3 的元素
    auto it = s.lower_bound(3);
    if (it != s.end()) {
        cout << "第一个大于等于 3 的元素是: " << *it << endl; // 输出 3
    }
}
```

```

    } else {
        cout << "没有大于等于 3 的元素" << endl;
    }

    return 0;
}

```

7, map

```

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

int main() {
    map<string, int> mp;

    // 插入键值对
    mp["apple"] = 5;    // 插入或修改键值对 "apple" -> 5
    mp["banana"] = 3;  // 插入 "banana" -> 3
    mp["orange"] = 7;  // 插入 "orange" -> 7

    // 判断键 "apple" 是否存在
    if (mp.count("apple") > 0) {
        cout << "键 'apple' 存在, 值为 " << mp["apple"] << endl;
    } else {
        cout << "键 'apple' 不存在" << endl;
    }

    // 遍历 map 中的所有键值对
    cout << "map 中的键值对为: " << endl;
    for (auto& p : mp) {
        cout << p.first << " -> " << p.second << endl;
    }

    return 0;
}

mp[1]=10;
mp[2]=20;
mp[3]=30;

mp.erase(2); // 删除 key=2

for(int i=0;i<n;i++){
    int x,y;
    cin >> x >> y; // 输入 key 和 value
    mp[x] = y;
}

```

```

}

for(auto x:mp) sum += x.second;//求和value

mp.size();//元素值

#include<bits/stdc++.h>
using namespace std;
#define ll long long
map<string,ll> mp;

int main()
{
    int q;
    cin >> q;

    ll sum = 0;                // ① 放到 while 外面，才能累计
    while(q-->0)
    {
        int x;
        cin >> x;

        if(x==1)
        {
            string s; ll y;
            cin >> s >> y;

            if(mp.find(s) == mp.end()) sum++;    // ② 只有新插入才+1
            mp[s] = y;

            cout << "OK" << endl;
        }

        if(x==2)
        {
            string s;
            cin >> s;

            auto it = mp.find(s);                // ③ 用 find 判断是否存在
            if(it == mp.end()) cout << "Not found" << endl;
            else cout << it->second << endl;
        }

        if(x==3)
        {
            string s;

```

```

    cin >> s;

    auto it = mp.find(s);           // ④ 用 find 判断是否存在
    if(it == mp.end()) cout << "Not found" << endl;
    else
    {
        mp.erase(it);             // ⑤ 真删除
        cout << "Deleted successfully" << endl;
        sum--;
    }
}

if(x==4)
{
    cout << sum << endl;
}
}
return 0;
}

```

二,蓝桥杯常用代码模板

1,快速排序 (直接用 STL)

```

#include <iostream>
#include <algorithm> // 包含sort函数
#include <vector>    // 包含vector定义
#include <functional> // 包含greater

using namespace std;

int main() {
    // 创建一个包含整数的vector
    int n;
    cin >> n;
    vector<int> v(n);

    for (int i = 0; i < n; i++) {
        cin >> v[i];
    }

    // 默认升序排序
    sort(v.begin(), v.end());
    cout << "升序排序: ";
    for (int num : v) {

```

```

        cout << num << " ";
    }
    cout << endl;
    // 降序排序
    sort(v.begin(), v.end(), greater<int>());
    cout << "降序排序: ";
    for (int num : v) {
        cout << num << " ";
    }
    cout << endl;

    return 0;
}
#include<bits/stdc++.h>
using namespace std;

int main()
{
    int n;
    cin >> n;

    int a[n];

    for(int i = 0; i < n; i++)
    {
        cin >> a[i];
    }

    sort(a, a + n, greater<int>()); // 降序排序

    for(int i = 0; i < n; i++)
    {
        cout << a[i] << " ";
    }

    cout << endl;

    return 0;
}

```

2,(1)二分查找

```

#include <iostream>
#include <vector>
#include <algorithm>

```

```

using namespace std;

int main() {
    // 假设有一个已排序的数组
    vector<int> v = {1, 2, 4, 6, 8, 10};
    int target = 5;

    // 使用 lower_bound 找到第一个 >= target 的位置
    int pos = lower_bound(v.begin(), v.end(), target) - v.begin();

    cout << "第一个 >= " << target << " 的位置是: " << pos << endl;
    return 0;
}

```

(2)二分答案

```

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

struct pos{
    int num;
    int sum;
}a[100005];

bool cmp(pos x,pos y){
    return x.sum < y.sum;
}

int main(){
    int n,m,q;
    cin>>n;

    for(int i=1;i<=n;i++){
        cin>>a[i].sum;
        a[i].num=i;
    }

    sort(a+1,a+1+n,cmp);

    cin>>m;

    for(int i=1;i<=m;i++){
        cin>>q;

        int l=0,r=n+1;

```

```

        while(l+1<r){
            int mid=(l+r)/2;
            if(a[mid].sum<=q)
                l=mid;
            else
                r=mid;
        }

        if(l>=1 && a[l].sum==q)
            cout<<a[l].num<<endl;
        else
            cout<<0<<endl;
    }
}

```

3,dfs

```

#include<bits/stdc++.h>
using namespace std;
#define int long long
#define endl '\n'
int n,m;
const int N=1e5+5;
int a[N];
bool st[N];
void dfs(int cnt)
{
    if(cnt>n)
    {
        for(int i=1;i<=n;i++)
        {
            cout<<a[i]<<" ";
        }
        cout<<endl;
        return ;
    }
    for(int i=1;i<=n;i++)
    {
        if(st[i]==0)
        {
            st[i]=1;
            a[cnt]=i;
            dfs(cnt+1);
            st[i]=0;
        }
    }
}

```

```

}
signed main()
{
    ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
cin>>n;
dfs(1);

    return 0;

}

```

```

#include<bits/stdc++.h>
using namespace std;
#define int long long
#define endl '\n'
int n,m;
const int N=1e5+5;
int a[N];
bool st[N];
void dfs(int cnt)
{
    if(cnt>n)
    {
        for(int i=1;i<=n;i++)
        {
            if(st[i])cout<<i<<" ";
        }
        cout<<endl;
        return ;
    }
    st[cnt]=0;
    dfs(cnt+1);
    st[cnt]=1;
    dfs(cnt+1);
}
signed main()
{
    ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
cin>>n;
dfs(1);
}

```

```
return 0;
```

```
}
```

```
#include<bits/stdc++.h>
using namespace std;
#define int long long
#define endl '\n'
int n,m;
const int N=1e5+5;
int a[N];
bool st[N];
void dfs(int cnt,int last)
{
    if(cnt>m)
    {
        for(int i=1;i<=m;i++)
        {
            cout<<a[i]<<" ";
        }
        cout<<endl;
        return ;
    }
    for(int i=last;i<=n;i++)
    {
        if(st[i]==0)
        {
            st[i]=1;
            a[cnt]=i;
            dfs(cnt+1,i);
            st[i]=0;
        }
    }
}
signed main()
{
    ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
    cin>>n>>m;
    dfs(1,1);

    return 0;
}
```

4,素数筛

```
#include<bits/stdc++.h>
using namespace std;
#define int long long
#define endl \n
int n,m;
const int N=1e6+5;
int a[N];
bool st[N];
signed main()
{
    ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
    cin>>n;
    int cnt=0;
    for(int i=2;i<=n;i++)
    {
        if(!st[i])
        {
            cnt++;
            for(int j=i*2;j<=n;j=j+i)
            {
                st[j]=1;
            }
        }
    }
    cout<<cnt;
    return 0;
}
```

5,最大公约数gcd和最小公倍数lcm

```
#include<bits/stdc++.h>
using namespace std;
#define int long long
#define endl \n
int a,b;
const int N=1e5+5;
int gcd(int a,int b)
{
    if(a==0)return b;
    return gcd(b%a,a);
}
```

```

int lcm(int a,int b)
{
    return a*b/gcd(a,b);
}
signed main()
{
    ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
    cin>>a>>b;
    cout<<lcm(a,b);
    return 0;
}

```

6,基础哈希表

```

#include<bits/stdc++.h>
using namespace std;
#define int long long
#define endl '\n'
int q;
unordered_set<int>jh;
signed main()
{
    ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
    cin>>q;
    while(q--)
    {
        char s;
        int x;
        cin>>s>>x;
        if(s=='I')
        {
            jh.insert(x);
        }
        else if(s=='Q')
        {
            if(jh.count(x))
            {
                cout<<"Yes"<<endl;
            }
            else cout<<"No"<<endl;
        }
    }
}

```

```
    return 0;
}
```

7,进制转换

```
#include<bits/stdc++.h>
using namespace std;
#define int long long
#define endl '\n'
int n,m;
const int N=1e5+5;
int a[N];
string s;
int k;
signed main()
{
    ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
    cin>>s>>k;
    int ans=0;
    for(int i=0;i<s.length();i++)
    {
        int x;
        if(s[i]>='0'&&s[i]<='9')
        {
            x=s[i]-'0';
        }
        else
        {
            x=s[i]-'A'+10;
        }
        ans=ans*k+x;
    }
    cout<<ans;
    return 0;
}
```

```
#include<bits/stdc++.h>
using namespace std;
#define int long long
#define endl '\n'
int n,m;
const int N=1e5+5;
```

```

int a[N];
int k;
signed main()
{
    ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
cin>>n>>k;
    int len=0;
    if(n == 0)
        {
            cout << 0;
            return 0;
        }
    while(n)
    {
        a[len++]=n%k;
        n/=k;
    }
for(int i=len-1;i>=0;i--)
{
    if(a[i]<10)cout<<a[i];
    else cout<<char('A'+a[i]-10);
}
    return 0;
}

```

8,结构体

```

#include<bits/stdc++.h>
using namespace std;
#define int long long
#define endl '\n'
int n,x,y;
const int N=200+5;
int a[N];
int b[N];
int id[N];
struct d
{
    int id;
    int len;
} f[N];
bool cmp(struct d p, struct d q)
{
    if(p.len!=q.len)return p.len < q.len;
}

```

```

        return p.id<q.id;
    }
signed main()
{
    ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
cin>>n>>x>>y;
for(int i=1;i<=n;i++)
{
    cin>>a[i]>>b[i];
}
for(int i=1;i<=n;i++)
{
    f[i].id=i;
    f[i].len=(x-a[i])*(x-a[i])+(y-b[i])*(y-b[i]);
}
sort(f+1,f+n+1,cmp);
for(int i=1;i<=3;i++)
{
    cout<<f[i].id<<endl;
}
    return 0;
}

```

9,分解质因数

```

#include<bits/stdc++.h>
using namespace std;
#define int long long
#define endl "\n"
int l,r;
const int N=1e5+5;
int a[N];
void check(int x)
{
    cout<<x<<"=";
    bool first=1;
    for(int i=2;i<=sqrt(x);i++)
    {
        while(x%i==0)
        {
            if(!first)cout<<"*";
            cout<<i;
            first=0;

```

```

        x/=i;
    }

}
if(x>1)
{
    if(!first)cout<<"*";
    cout<<x;
}
    cout<<endl;
}
signed main()
{
    ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
    cin>>l>>r;
    for(int i=l;i<=r;i++)
    {
        check(i);
    }

    return 0;
}

```

10结构体

```

#include<bits/stdc++.h>
using namespace std;
#define int long long
#define endl '\n'
int n,x,y;
const int N=200+5;
int a[N];
int b[N];
int id[N];
struct d
{
    int id;
    int len;
} f[N];
bool cmp(struct d p, struct d q)
{
    if(p.len!=q.len)return p.len < q.len;
    return p.id<q.id;
}

```

```

signed main()
{
    ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
    cin>>n>>x>>y;
    for(int i=1;i<=n;i++)
    {
        cin>>a[i]>>b[i];
    }
    for(int i=1;i<=n;i++)
    {
        f[i].id=i;
        f[i].len=(x-a[i])*(x-a[i])+(y-b[i])*(y-b[i]);
    }
    sort(f+1,f+n+1,cmp);
    for(int i=1;i<=3;i++)
    {
        cout<<f[i].id<<endl;
    }
    return 0;
}

```

11,模板

```

#include<bits/stdc++.h>
using namespace std;
#define int long long
#define endl '\n'
int n,m;
const int N=1e5+5;
int a[N];

signed main()
{
    ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);

    cin>>n;
    cout<<n;
}

```

```
return 0;  
}
```