

青少年软件编程（C 语言）等级考试试卷（九级）

分数：100 题数：4

一、编程题(共 4 题，共 100 分)

1. 数对

题目描述

小 A 有 n 个数对 (a_i, b_i) 。不保证这些数对互不相同。

对于所有满足 $1 \leq i < j \leq n$ 的正整数 i, j ，小 A 认为数对 (a_i, b_i) 和数对 (a_j, b_j) 是好朋友，当且仅当 $a_i \neq a_j$ 且 $b_i \neq b_j$ 。

小 A 需要你求出，这 n 个数对之间有多少对好朋友。

输入格式

第一行包含一个整数 n 。

接下来 n 行，第 i 行包含两个整数 a_i, b_i 。

输出格式

输出一行，包含一个整数，表示这 n 个数对之间的好朋友数量。

输入样例

```
4
1 2
1 3
2 3
2 4
```

输出样例

说明提示

样例 1 解释

是好朋友的数对有：

- 数对 $(1, 2)$ 和数对 $(2, 3)$ ；
- 数对 $(1, 2)$ 和数对 $(2, 4)$ ；
- 数对 $(1, 3)$ 和数对 $(2, 4)$ ；

一共有 3 对好朋友。

样例 2

见 *pair/pair2.in* 与 *pair/pair2.ans*。

该组样例满足测试点 2 的限制。

样例 3

见 *pair/pair3.in* 与 *pair/pair3.ans*。

该组样例满足测试点 3 的限制。

样例 4

见 *pair/pair4.in* 与 *pair/pair4.ans*。

该组样例满足测试点 5 的限制。

数据范围

对于所有测试数据，保证：

- $1 \leq n \leq 5 \times 10^5$ ；
- $1 \leq a_i, b_i \leq 10^9$ 。

测试点编号	$n \leq$	特殊性质
1	10^3	是
2	10^3	否
3 ~ 4	5×10^5	是
5	5×10^5	否

特殊性质：保证所有数对互不相同。

限制

时间限制：2000ms

内存限制：512MiB

试题编号：20250906-9-01

试题类型：编程题

标准答案：

试题难度：一般

试题解析：

展示地址：[点击浏览](#)

参考代码：

```
#include<bits/stdc++.h>
using namespace std;
map<int,int> mp1,mp2;
map<pair<int,int>,int> mp3;
int main(){
    ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
    // freopen("pair.in","r",stdin);
```

```
// freopen("pair.out","w",stdout);
long long n;
cin>>n;
long long ans=n*(n-1)/2;
for (int i=1;i<=n;i++){
    int x,y;
    cin>>x>>y;
    ans-=mp1[x]+mp2[y];
    ans+=mp3[make_pair(x,y)];
    mp1[x]++;
    mp2[y]++;
    mp3[make_pair(x,y)]++;
}
cout<<ans;
return 0;
}
```

考生得分 : 0

是否评分 : 已评分

评价描述 :

2. 序列

题目描述

小 B 有一个长度为 n 的序列 v 。

他定义一次操作为：

- 设序列 v 末尾的元素为 x ；
- 计算当前序列 v 中所有元素之和 s ；
- 将 $s + x$ 加入到序列 v 的末尾。

小 B 需要你求出，进行 m 次操作后，序列 v 中所有元素之和为多少。

由于答案可能很大，你只需要求出答案对 998244353 取模后的结果。

输入格式

第一行包含两个整数 n, m 。

第二行包含 n 个整数 $v_1, \dots, v_n$ 。

输出格式

输出一行，包含一个整数，表示进行 m 次操作后序列 v 中所有元素之和对 998244353 取模后的结果。

输入样例

```
2 1
3 5
```

输出样例

```
21
```

说明提示

样例 1 解释

$(3 + 5) + 5 = 13$ ，所以进行 m 次操作后序列 v 中所有元素之和为 $3 + 5 + 13$ ，即 21。

样例 2

见 *sequence/sequence2.in* 与 *sequence/sequence2.ans*。

该组样例满足测试点 2 的限制。

样例 3

见 *sequence/sequence3.in* 与 *sequence/sequence3.ans*。

该组样例满足测试点 4 的限制。

样例 4

见 *sequence/sequence4.in* 与 *sequence/sequence4.ans*。

该组样例满足测试点 5 的限制。

数据范围

对于所有测试数据，保证：

- $1 \leq n \leq 10^6, 1 \leq m \leq 10^{18}$ ；
- $1 \leq v_i \leq 10^9$ 。

测试点编号	$n \leq$	$m \leq$
1	1	10^6
2 ~ 3	10^6	10^6
4	1	10^{18}
5	10^6	10^{18}

试题编号：20250906-9-02

试题类型：编程题

标准答案：

试题难度：一般

试题解析：

展示地址：[点击浏览](#)

参考代码：

```
#include <iostream>
```

```
#include <vector>
```

```
using namespace std;
```

```
using LL = long long;
```

```
const int kN = 1e6 + 5;
```

```
const int Mod = 998244353;
```

```
int n;
```

```
LL m;
```

```
int a[kN];
```

```
struct matrix {
```

```
    int a[2][2];
```

```
    int* operator[] (int x) {
```

```
        return a[x];
```

```
    }
```

```

friend matrix operator* (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 i = 0; i < 2; ++i) {
        for (int j = 0; j < 2; ++j) {
            for (int k = 0; k < 2; ++k) {
                c[i][j] = (c[i][j] + 1ll * a[i][k] * b[k][j]) % Mod;
            }
        }
    }
    return c;
}

```

```

friend vector<int> operator* (matrix a, vector<int> b) {
    vector<int> c(2);
    for (int i = 0; i < 2; ++i) {
        c[i] = 0;
        for (int j = 0; j < 2; ++j) {
            c[i] = (c[i] + 1ll * a[i][j] * b[j]) % Mod;
        }
    }
    return c;
}
};

```

```

matrix qpow (matrix x, LL y) {
    matrix r;
    r[0][0] = 1, r[0][1] = 0;
    r[1][0] = 0, r[1][1] = 1;
    for (; y; y >>= 1) {
        if (y & 1) r = r * x;
        x = x * x;
    }
    return r;
}

```

```

void fakemain () {
    cin >> n >> m;
    int sum = 0;
    for (int i = 1; i <= n; ++i) {

```

```

        cin >> a[i];
        sum = (sum + a[i]) % Mod;
    }
    matrix mat;
    mat[0][0] = 1, mat[0][1] = 1;
    mat[1][0] = 1, mat[1][1] = 2;
    mat = qpow(mat, m);
    vector<int> vec{a[n], sum};
    vec = mat * vec;
    cout << vec[1] << '\n';
}

int main () {
    // freopen("sequence.in", "r", stdin);
    // freopen("sequence.out", "w", stdout);
    cin.tie(0)->sync_with_stdio(0);
    fakemain();
    return 0;
}

```

考生得分 : 15

是否评分 : 已评分

评价描述 :

3. 概率

题目描述

小 C 有 n 个骰子，第 i 个骰子的面数为 a_i ，第 j 个面上的点数为 j 。

现在，他会同时投掷这 n 个骰子。对于第 i 个骰子，其每个点数被掷出的概率相同，均为 $\frac{1}{a_i}$ 。

小 C 需要你求出，这 n 个骰子掷出的点数均不同的概率对 $10^9 + 7$ 取模后的结果。

输入格式

第一行包含一个整数 n 。

第二行包含 n 个整数 $a_1, \dots, a_n$ 。

输出格式

输出一行，包含一个整数，表示这 n 个骰子掷出的点数均不同的概率对 $10^9 + 7$ 取模后的结果。即设最终答案为 $\frac{p}{q}$ ，则需要输出 $p \times q^{10^9+5}$ 对 $10^9 + 7$ 取模后的结果。

输入样例#1

```
2
3 5
```

输出样例#1

```
600000005
```

输入样例#2

```
3
2 2 2
```

输出样例#2

```
0
```

说明提示

样例 1 解释

只有当两个骰子都掷出 1 或 2 或 3 时才不满足条件，所以这两个骰子掷出的点数均不同的概率为 $\frac{4}{5}$ ，而 $\frac{4}{5} \equiv 600000005 \pmod{10^9 + 7}$ ，所以答案为 600000005。

样例 2 解释

显然无法做到这三个骰子掷出的点数均不同，所以答案为 0。

样例 3

见 *probability/probability3.in* 与 *probability/probability3.ans*。

该组样例满足测试点 3 的限制。

样例 4

见 *probability/probability4.in* 与 *probability/probability4.ans*。

该组样例满足测试点 5 的限制。

样例 5

见 *probability/probability5.in* 与 *probability/probability5.ans*。

该组样例满足测试点 10 的限制。

数据范围

对于所有测试数据，保证：

- $1 \leq n \leq 10^6$;
- $1 \leq a_i \leq 10^9$ 。

测试点编号	$n \leq$	$a_i \leq$	特殊性质
1 ~ 2	2	10^9	无
3 ~ 4	8	8	无
5	10^6	10^9	A
6 ~ 7	10^6	10^9	B
8 ~ 10	10^6	10^9	无

- 特殊性质 A：保证 $a_i = i$ 。
- 特殊性质 B：保证所有 a_i 均相等。

限制

时间限制：1000ms

内存限制：512MiB

试题编号：20250906-9-03

试题类型：编程题

标准答案：

试题难度：一般

试题解析：

展示地址：点击浏览

参考代码：

```
#include <bits/stdc++.h>
using namespace std;
const int N=1e6+5,mod=1e9+7;
int n,a[N];
int ksm(int a,int b){
    int res=1;
    while(b){
        if(b&1) res=1ll*res*a%mod;
        a=1ll*a*a%mod;
        b>>=1;
    }
}
```

```

    }
    return res;
}
int main(){
//  freopen("probability.in","r",stdin);
//  freopen("probability.out","w",stdout);
    cin>>n;
    for(int i=1;i<=n;i++) cin>>a[i];
    sort(a+1,a+1+n);
    int ans=1;
    for(int i=1;i<=n;i++){
        if(a[i]<i){
            cout<<0<<endl;
            return 0;
        }
        ans=1ll*ans*ksm(a[i],mod-2)%mod*(a[i]-i+1)%mod;
    }
    cout<<ans;
    return 0;
}

```

考生得分：25

是否评分：已评分

评价描述：

4. 游戏

题目描述

小 D 正在玩一个游戏。

这个游戏在一个包含 n 个结点和 m 条无向边的无向连通图上进行，游戏规则如下：

- 初始时所有结点都为白色；
- 系统会随机选择一个结点 u 作为初始结点；
- 小 D 需要指定一个不大于 n 的正整数 k ，系统会随机 k 个结点的颜色变为黑色；
- 在小 D 指定正整数 k 后，系统会随机一条边切断；
- 如果在边被切断后，存在一个黑色结点 v 满足 $u = v$ 或结点 u 可以经过未被切断的边到达结点 v ，则小 D 获胜，否则小 D 失败。

小 D 需要你求出，对于每个初始结点 u ，小 D 所指定的正整数 k 至少要为多少，才能满足小 D 一定可以获胜。

输入格式

第一行包含两个整数 n, m 。

接下来 m 行，第 i 行包含两个整数 u_i, v_i ，表示第 i 条边连接结点 u_i 和结点 v_i 。

输出格式

输出一行，包含 n 个整数，第 i 个整数表示 $u = i$ 时使小 D 一定可以获胜所需要指定的正整数 k 的最小值。

输入样例

```
3 2
2 1
2 3
```

输出样例

```
3 2 3
```

说明提示

样例 1 解释

当 $u = 1$ 时，小 D 所指定的正整数 k 至少要为 3 才能满足小 D 一定可以获胜；因为当 $k = 2$ 时，如果系统将第 2 个结点和第 3 个结点的颜色变为黑色并将第 1 条边切断，则小 D 会失败，而当 $k = 3$ 时小 D 一定可以获胜。

样例 2

见 *game/game2.in* 与 *game/game2.ans*。

该组样例满足测试点 6 的限制。

样例 3

见 *game/game3.in* 与 *game/game3.ans*。

该组样例满足测试点 9 的限制。

样例 4

见 *game/game4.in* 与 *game/game4.ans*。

该组样例满足测试点 11 的限制。

样例 5

见 *game/game5.in* 与 *game/game5.ans*。

该组样例满足测试点 13 的限制。

样例 6

见 *game/game6.in* 与 *game/game6.ans*。

该组样例满足测试点 16 的限制。

样例 7

见 *game/game7.in* 与 *game/game7.ans*。

该组样例满足测试点 20 的限制。

数据范围

对于所有测试数据，保证：

- $2 \leq n \leq 3 \times 10^5, n - 1 \leq m \leq 5 \times 10^5$;
- $1 \leq u_i, v_i \leq n, u_i \neq v_i$;
- 不存在 $1 \leq i, j \leq m$, 满足 $i \neq j$ 且 $u_i = u_j$ 且 $v_i = v_j$;
- 不存在 $1 \leq i, j \leq m$, 满足 $i \neq j$ 且 $u_i = v_j$ 且 $v_i = u_j$;
- 所有 (u_i, v_i) 形成一个连通图。

测试点编号	$n \leq$	$m \leq$
1 ~ 2	10	$n - 1$
3 ~ 4	10	20
5	80	$n - 1$
6	80	100
7 ~ 8	300	$n - 1$
9 ~ 10	300	500
11 ~ 12	3×10^3	$n - 1$
13 ~ 15	3×10^3	5×10^3
16 ~ 17	3×10^5	$n - 1$
18 ~ 20	3×10^5	5×10^5

限制

时间限制：1000ms

内存限制：512MiB

试题编号：20250906-9-04

试题类型：编程题

标准答案：

试题难度：一般

试题解析：

展示地址：点击浏览

参考代码：

```
#include<iostream>
```

```

#include<algorithm>
#include<vector>
#include<stack>
const int sz=3e5+10;
struct edge{
    int nxt,to;
}graph[sz<<3];
int head[sz],hpp=1;
void addEdge(int from,int to){
    graph[++hpp]=edge{head[from],to};
    head[from]=hpp;
}
int dfn[sz],low[sz],dpp,edcc,belong[sz];
std::vector<int>points[sz];
std::stack<int>s;
void tarjan(int u,int pid){
    dfn[u]=low[u]=++dpp;
    s.push(u);
    for(int p=head[u];p;p=graph[p].nxt){
        int v=graph[p].to;
        if(dfn[v]==0)tarjan(v,p),low[u]=std::min(low[u],low[v]);
        else if((p^1)!=pid)low[u]=std::min(low[u],dfn[v]);
    }
    if(dfn[u]==low[u]){
        ++edcc;
        while(!s.empty()){
            int v=s.top();
            s.pop();
            points[edcc].push_back(v);
            belong[v]=edcc;
            if(v==u)break;
        }
    }
}
std::vector<int>tree[sz];
int size[sz],max[sz];
void dfs(int u,int fau){
    size[u]=points[u].size(),max[u]=0;
    for(int v:tree[u]){
        if(v==fau)continue;
        dfs(v,u);
        size[u]+=size[v];
        max[u]=std::max(max[u],size[v]);
    }
}

```

```

}
int main(){
    // freopen("game.in","r",stdin);
    // freopen("game.out","w",stdout);
    int n,m;
    std::cin>>n>>m;
    std::fill(head+1,head+n+1,0),hpp=1;
    for(int i=1,u,v;i<=m;i++){
        std::cin>>u>>v,addEdge(u,v),addEdge(v,u);
    }
    std::fill(dfn+1,dfn+n+1,0);
    for(int i=1;i<=edcc;i++)tree[i].clear(),points[i].clear();
    edcc=0;
    for(int u=1;u<=n;u++){
        if(dfn[u]==0)tarjan(u,0);
    }
    for(int u=1;u<=n;u++){
        for(int p=head[u];p;p=graph[p].nxt){
            int v=graph[p].to;
            if(belong[u]!=belong[v])tree[belong[u]].push_back(belong[v]);
        }
    }
    dfs(1,0);
    for(int i=1;i<=n;i++)std::cout<<std::max(max[belong[i]],n-size[belong[i]])
+1<<" ";
    std::cout<<"\n";
    return 0;
}

```

考生得分：13

是否评分：已评分

评价描述：