

## 基础

离散化nlogn

文件读写方式

## int128

快读 快写 //c++17不能用register int x int st[33]

二分 三分

二分

三分

前缀差分

二维前缀

排序

并归排序

快排

结构体内部+sort

高精度acwing

简易大数处理

# 基础

## 离散化nlogn

```
scanf("%d",&n);
for(int i=1;i<=n;i++)
{
    scanf("%d",&a[++tot]);    //读个入
    b[tot]=a[tot];
}
sort(a+1,a+1+n);            //排个序
tot=unique(a+1,a+1+tot)-a-1;    //去个重
for(int i=1;i<=n;i++)b[i]=lower_bound(a+1,a+1+tot,b[i])-a;//离散化一下
for(int i=1;i<=n;i++)
{
    add(b[i],1);            //动态加点
    if(i&1)printf("%d\n",a[find_kth((i+1)>>1)]);//查kth
}
}
```

```
*****
int a[N];
int b[N];
int n,cnt=0;
cin>>n;
for(int i=1;i<=n;i++)cin>>a[i],b[cnt++]=a[i];
for(int i=1;i<=n;i++)
{
    cout<<a[i]<<" ";
}
}
```

```

cout<<endl;
sort(b,b+cnt);
cnt=unique(b,b+cnt)-b;
for(int i=1;i<=n;i++)a[i]=lower_bound(b,b+n,a[i])-b,a[i]++;//离散化处理

for(int i=1;i<=n;i++)
{
    cout<<a[i]<<" ";
}

```

```

int a[M],d[M],t[M],n;
//原数组/ 离散化后的数组/ 树状数组
bool cmp(int x,int y)
{
    if(a[x]==a[y]) return x>y;//避免元素相同
    return a[x]>a[y];//按照原序列第几大排列
}
int main()
{
    cin>>n;
    for(int i=1;i<=n;i++)
        cin>>a[i],d[i]=i;//初始化

    sort(d+1,d+n+1,cmp);
    //排序时候d就是离散化的数组了

    return 0;
}

```

版权声明：本文为CSDN博主「baby的我」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：[https://blog.csdn.net/qq\\_41431457/article/details/88945833](https://blog.csdn.net/qq_41431457/article/details/88945833)

## 文件读写方式

```

freopen("input.txt","r",stdin);
freopen("output.txt","w",stdout);

```

## int128

```

inline __int128 read()
{
    __int128 x=0,f=1;
    char c=getchar();
    while(c<'0' || c>'9')
    {
        if(c=='-') f=-1;
        c=getchar();
    }

```

```

    }
    while(c>='0'&& c<='9')
    {
        x=(x<<3)+(x<<1)+(c^48);
        c=getchar();
    }
    return x*f;
}

void write(__int128 x){
    if(x>9) write(x/10);
    putchar(x%10^48);
}

```

## 快读 快写 //c++17不能用register int x int st[33]

```

#include<bits/stdc++.h>
using namespace std;
#define ll long long
const int inf=0x3f3f3f3f;
inline int read()
{
    int x=0,f=1;
    char c=getchar();
    while(c<'0' || c>'9')
    {
        if(c=='-') f=-1;
        c=getchar();
    }
    while(c>='0'&& c<='9')
    {
        x=(x<<3)+(x<<1)+(c^48);
        c=getchar();
    }
    return x*f;
}

void write(int x){
    if(x>9) write(x/10);
    putchar(x%10^48);
}

inline void write( int x) {
    if (x == 0) putchar('0');
    else {
        int st[33], tp = 0;
        while (st[++tp] = x % 10, x /= 10);
        while (putchar(st[tp] ^ 48), --tp);
    }
    putchar('\n');
}

int main()

```

```

{
    ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
    int a;
    return 0;
}

```

## 二分 三分

### 二分

```

#include <bits/stdc++.h>
using namespace std;
#define ll long long
const int inf=0x3f3f3f3f;
const int N=1e5+1;
int a[N];

int main()
{
    ios::sync_with_stdio(false);
    int n,q;
    cin>>n>>q;
    for(int i=1;i<=n;i++)
    {
        cin>>a[i];
    }
    for(int i=1;i<=q;i++)
    {
        int x;
        cin>>x;
        int l=1;int r=n;
        int mid=(l+r)/2;
        while(l<r)
        {
            mid=(l+r)/2;
            if(a[mid]>=x)
            {
                r=mid;
            }
            else
            {
                l=mid+1;
            }
        }
        int temp1=l-1;
        //cout<<l-1<<" ";
    }
}

```

```

l=1;r=n;
while(l<r)
{
    mid=(l+r+1)/2;
    if(a[mid]<=x)
    {
        l=mid;
    }
    else
    {
        r=mid-1;
    }
}
int temp2=l-1;
//cout<<l-1<<endl;
if(a[temp1+1]==x&&a[temp2+1]==x)
{
    cout<<temp1<<" "<<temp2<<"\n";
}
else cout<<-1<<" "<<-1<<"\n";
}
return 0;
}

```

## 三分

```

const double eps=1e-8;
double Ax,Ay,Bx,By,Cx,Cy,Dx,Dy;
double p,q,r;
double len(double x1,double y1,double x2,double y2)
{
    return sqrt(pow(x1-x2,2)+pow(y1-y2,2));
}
double time_EFD(double Ex,double Ey)//三分点F;
{
    double lx=Cx,rx=Dx,ly=Cy,ry=Dy;
    while(len(lx,ly,rx,ry)>eps)
    {
        double lmidx=lx+(rx-lx)/3;
        double lmidy=ly+(ry-ly)/3;
        double rmidx=rx-(rx-lx)/3;
        double rmidy=ry-(ry-ly)/3;
        double time1=len(Ex,Ey,lmidx,lmidy)/r+len(lmidx,lmidy,Dx,Dy)/q;
        double time2=len(Ex,Ey,rmidx,rmidy)/r+len(rmidx,rmidy,Dx,Dy)/q;
        if(time1-time2>eps)
        {
            lx=lmidx;
            ly=lmidy;
        }
    }
}

```

```

        else
        {
            rx=rmidx;
            ry=rmidy;
        }
    }
    return len(Ex,Ey,lx,ly)/r+len(lx,ly,Dx,Dy)/q;//time = EF/r + FD/q;
}
double time_all()//三分E点
{
    double lx=Ax,ly=Ay;
    double rx=Bx,ry=By;
    while(len(lx,ly,rx,ry)>eps)
    {
        double lmidx=lx+(rx-lx)/3;
        double lmidy=ly+(ry-ly)/3;
        double rmidx=rx-(rx-lx)/3;
        double rmidy=ry-(ry-ly)/3;
        double time1=len(Ax,Ay,lmidx,lmidy)/p+time_EFD(lmidx,lmidy);
        double time2=len(Ax,Ay,rmidx,rmidy)/p+time_EFD(rmidx,rmidy);
        if(time1-time2>eps)
        {
            lx=lmidx;
            ly=lmidy;
        }
        else
        {
            rx=rmidx;
            ry=rmidy;
        }
    }
    return len(Ax,Ay,lx,ly)/p+time_EFD(lx,ly);
}
int main()
{
    cin>>Ax>>Ay>>Bx>>By>>Cx>>Cy>>Dx>>Dy;
    cin>>p>>q>>r;
    printf("%.2f\n",time_all());
    return 0;
}

```

## 前缀差分

### 二维前缀

```

#include<bits/stdc++.h>
using namespace std;
int main()
{
    int sum[200][200];
    int qz[200][200];

```

```

int n;
cin>>n;
int a[200][200];
for(int i=1;i<=n;i++)
{
    for(int j=1;j<=n;j++)
    {
        cin>>a[i][j];
        qz[i][j]=qz[i][j-1]+a[i][j];
        sum[i][j]=sum[i-1][j]+qz[i][j];
    }
}
int temp;
int maxx=-2147483648;
for(int x1=1;x1<=n;x1++)
{
    for(int x2=1;x2<=n;x2++)
    {
        for(int y1=1;y1<=n;y1++)
        {
            for(int y2=1;y2<=n;y2++)
            {
                if(x1>x2||y1>y2)continue;
                else
                {
                    temp=sum[x2][y2]-sum[x1-1][y2]-sum[x2][y1-1]+sum[x1-1][y1-
1];

                    maxx=max(maxx,temp);
                }
            }
        }
    }
}
cout<<maxx;
return 0;
}

```

## 排序

### 并归排序

```

#include<bits/stdc++.h>
using namespace std;
int ans;
int a[1000000];
int c[1000000];
void msort(int b,int e)
{
    if(b==e)return ;
    int mid=b+e>>1;
    int i=b;int j=mid+1;int k=b;

```

```

msort(b,mid);
msort(mid+1,e);
while(i<=mid&& j<=e)
{
    if(a[i]<=a[j])
        c[k++]=a[i++];
    else
    {
        c[k++]=a[j++];
        ans=ans+mid+1-i;
    }
}
//有剩余，补充上
while(i<=mid)
{
    c[k++]=a[i++];
}
while(j<=e)
{
    c[k++]=a[j++];
}
//更新序列;
for(int v=b;v<=e;v++)
    a[v]=c[v];
}
int main()
{
    int n;
    cin>>n;
    for(int i=1;i<=n;i++)
    {
        cin>>a[i];
    }
    msort(1,n);
    for(int i=1;i<=n;i++)
    {
        cout<<a[i]<<" ";
    }
    cout<<endl<<ans;
    return 0;
}

```

## 快排

```

#include <bits/stdc++.h>
using namespace std;
const int N=1e6+10;
int a[N];
void qsort(int l,int r)
{

```



```

    if(l>=r)return ;
    int x=a[l+r>>1];
    int i=l-1;int j=r+1;
    while(i<j)
    {
        do i++; while(a[i]<x);
        do j--; while(a[j]>x);
        if(i<j)swap(a[i],a[j]);
    }
    qsort(l,j);
    qsort(j+1,r);
}
int main()
{
    int n;
    cin>>n;
    for(int i=1;i<=n;i++)
    {
        cin>>a[i];
    }
    qsort(1,n);
    for(int i=1;i<=n;i++)
    {
        cout<<a[i]<<" ";
    }
    return 0;
}

```

## 结构体内部+sort

```

struct node
{
    int l,r;
    bool operator <(const node &a)const{
        return r < a.r;
    }
};

```

## 高精度acwing

```

#include <bits/stdc++.h>
using namespace std;
#define ll long long
const ll inf=0x3f3f3f3f;
const ll N=1e5+1;
vector<int> add(vector<int> &A, vector<int> &B)
{
    if (A.size() < B.size()) return add(B, A);

    vector<int> C;

```

```

    int t = 0;
    for (int i = 0; i < A.size(); i ++ )
    {
        t += A[i];
        if (i < B.size()) t += B[i];
        C.push_back(t % 10);
        t /= 10;
    }

    if (t) C.push_back(t);
    return C;
}

vector<int> sub(vector<int> &A, vector<int> &B)
{
    vector<int> C;
    for (int i = 0, t = 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;
}

vector<int> mul(vector<int> &A, int b)
{
    vector<int> C;

    int t = 0;
    for (int i = 0; i < A.size() || t; i ++ )
    {
        if (i < A.size()) t += A[i] * b;
        C.push_back(t % 10);
        t /= 10;
    }

    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;
    }
    reverse(C.begin(), C.end());
    while (C.size() > 1 && C.back() == 0) C.pop_back();
    return C;
}
//add
int main()
{
    ios::sync_with_stdio(false);
    string a,b;
    vector<int>A,B;
    cin>>a>>b;
    for(int i=a.size()-1;i>=0;i--)A.emplace_back(a[i]-'0');
    for(int i=b.size()-1;i>=0;i--)B.emplace_back(b[i]-'0');
    auto C=add(A,B);
    for(int i=C.size()-1;i>=0;i--)cout<<C[i];
    return 0;
}
//sub
int main()
{
    ios::sync_with_stdio(false);
    string a,b;
    vector<int>A,B;
    cin>>a>>b;
    for(int i=a.size()-1;i>=0;i--)A.emplace_back(a[i]-'0');
    for(int i=b.size()-1;i>=0;i--)B.emplace_back(b[i]-'0');
    vector<int>C;

    if(a.size()<b.size()){cout<<"-";swap(A,B);}
    else if(a.size()==b.size())
    {
        if(a<b)
        {
            cout<<"-";
            swap(A,B);
        }
    }
    C=sub(A,B);
    for(int i=C.size()-1;i>=0;i--)cout<<C[i];
    return 0;
}
int main()
{
    ios::sync_with_stdio(false);
    string a;int b;
    vector<int>A,B;
    cin>>a>>b;
    for(int i=a.size()-1;i>=0;i--)A.emplace_back(a[i]-'0');
    auto C=mul(A,b);
    for(int i=C.size()-1;i>=0;i--)cout<<C[i];

```

```

    return 0;
}
int main()
{
    string a;
    vector<int> A;
    int b;
    cin>>a>>b;
    for (int i=a.size()-1;i>=0;i--)A.push_back(a[i]-'0');
    int r;
    auto C =div(A, b, r);
    for(int i=C.size()-1;i>=0;i--)
        cout<<C[i];
    cout<<endl<<r<<endl;
    return 0;
}

```

## 简易大数处理

统计答案时，我们可以用两个数来存，先用第一个数存答案，当答案大小超出 $1e18$ 时，我们就将超出部分存到第二个数，这里的超出部分是指第 $1e18$ 位之后的数，也就是这两位数拼起来就是答案  
这部分对应的代码：

```

ll te=1e18
if(ans[0]>=te)ans[1]+=ans[0]/te,ans[0]%=te;

if(ans[1])printf("%lld%018lld\n",ans[1],ans[0]);
else printf("%lld\n",ans[0]);

```