

# 20221114 测试总结

只能拿一些部分分😭

	T1	T2	T3	T4	总分
估分	0	20	0	有分	20-100
得分	0	20	0	42	62

## T1 最短母串(substr)

这道题先拿到的时候，看到是字符串有些慌。看到是有关相同串的问题，就想到了KMP，试图将KMP套上去。当时想的是枚举每一个字符串同时记录一个字符串作为答案，每次都去用KMP找相同的串，如果不一样就在ans串上增加。但是这样大概率会WA（可是当时想的是有答案输出就行，万一能骗点分呢），结果发现KMP套不上去，只能贪心（输出最长的字符串，果然没有分）。隔壁Sin巨用的特判+最长相同子链拿了20分，%%%。

后面看题解是用的类似于AC自动机的思路，建立fail树，以及在字符串的结尾用2的幂次来记录是否找到了对应的母串（感谢孙巨给我们讲AC自动机和这道题的思路💡）。

## T2 要养花(flower)

一个分块题。考试的时候，看到后面的有部分分就直接先打了个暴力拿了20分。后面想的是求得是一个区间内的最大值，想到了线段树，但是是求 $\%k$ 意义下的区间最大值，用线段树的话就没有想到建立的方法。于是又想到离线处理，把每个 $k$ 存下来，然后每一个 $k$ 建立一个线段树，发现这样写，不仅会MLE还会TLE，不如打暴力，于是就卡卡常就走了。

改题的时候才知道这道题其实是一个分块题。以前没有怎么写过分块的题，于是第二道改这道题（再次感谢孙巨和Siri💡）。这道题首先确定块的大小，然后先预处理每一个块，求出一个qwq数组（胖巨和Sin巨总是吐槽我的这个数组名），里面存的是对于每一个块而言，小于等于当前值的最大值，纠结了挺久了才懂这一步。后面就要好理解一些，就是找每一块中 $\%k$ 意义下的最大值（还要枚举 $k$ 的倍数）。最后询问的时候就可以先分类讨论一下，这里调了一会儿，但是自己手推一下就差不多能写对询问时对应块的编号。这道题改了之后还是很有收获的。

## T3 折射(refract)

这道题其实跟光学一点关系都没有，当时做题的时候先是读了好一会儿题才理解，就想着打dfs，但是不仅WA了，复杂度还挺高（毕竟样例都推错了）。后面就开始乱搞，先是排序，换了好几次排序方式，但是都没有找到一个好一点的排序方式。于是开始骗分（能过样例就行），果然没有分。

下午改题的时候先改的这道题，就花费了我们一下午的时间，问了好多个人（问了四个人左右，很谢谢他们，都讲了好多遍，每一遍都很认真，💡）。这道题是dp（当初真没有看出来），题解写得很简略（所以我们才去问了好多遍），但确实把主要思路写出来了。先是定义 $f[i][0/1]$ 表示以 $i$ 为端点，0、1表示先向右或左转的方案数。通过多次画图发现其实光线每次都交替转折，将装置按照 $x$ 坐标从小到大排序递推求取，再用前缀和优化（就是这个地方听了很多遍），这样的话dp方程就大概出来了，写出来的代码其实很短。当天就只改了这一道题，晚上只去问了第二题（但是还没有完全没有听懂，后面又去问）。

## T4 画作(paint)

这道题适合打搜索骗分（搜索果然好用）。考试的时候看到的时候觉得跟之前的寒假做的细胞很像，于是就打了个dfs。然后发现如果只判断四联通的块数会WA，但是判断八联通也会WA，于是写了两种情况的判断。如果是八联通块的话ans++。结果写完的时候发现还是在搜索的时候有问题，但是改不出来。后面写了两个dfs再特判了一下，把样例骗过了。最后拿了42分，挺惊喜的。后面想到这道题打bfs其实更优一些，会避免一些应该先找四联通块再找八联通块却找反的情况。下午看分的时候，隔壁Sin巨打的bfs拿了56分，%%%。

## 总结：

几乎每道都打的是部分分的骗分，有些骗到了，有些没有骗到，代码能力还是要加强。但是从今天的题看来，会打搜索真的很重要，是个骗分的重要技巧。但是dp方面的能力还是欠缺，总是想不到这道题可以用dp（虽然想到也有可能推不出来转移方程）。

最后挂一挂改对的代码：

```
//要养花
#include<cstdio>
#include<cstring>
#include<cctype>
#include<iostream>
#include<bitset>
using namespace std;
const int MAXN = 100010;
const int B = 1000;

template<typename T>inline void read(T &x)
{
    int c=getchar();
    x=0;
    T neg=0;
    while(!isdigit(c)) neg|=(c^'-'),c=getchar();
    while(isdigit(c)) x=(x<<1)+(x<<3)+(c^48),c=getchar();
    if(neg) x=(~x)+1;
}

int n,m,a[MAXN],maxn,b,qwq[MAXN],ans[110][MAXN];

inline int maxa(int x,int y)
{
    return x>y ? x : y;
}

inline int mina(int x,int y)
{
    return x>y ? y : x;
}

int main()
{
    freopen("flower.in","r",stdin);
    freopen("flower.out","w",stdout);
    read(n),read(m);
    register int i,j,k;
    int l,res,x,y,r;
    for(i=1;i<=n;++i)
        read(a[i]);
    b=(n-1)/B+1;
```

```

for(i=1;i<=b;++i)
{
    memset(qwq,0,sizeof(qwq));
    for(j=(i-1)*B+1;j<=i*B&& j<=n;++j)
        qwq[a[j]]=a[j];
    for(j=1;j<=MAXN-10;++j)
        qwq[j]=maxa(qwq[j],qwq[j-1]);
    for(j=2;j<=(MAXN-10);++j)
        for(k=0;k<=(MAXN-10);k+=j)
        {
            ans[i][j]=maxa(ans[i][j],qwq[mina(k+j-1,MAXN-10)]-k);
//            printf("%d %d %d\n",i,j,ans[i][j]);
        }

}
// for(j=1;j<=10;++j)
//     printf("%d %d\n",j,ans[1][j]);
while(m--)
{
    read(l),read(r),read(k);
    x=l/B+1,y=r/B,maxn=0;
    if(x==y+1)
    {
        for(i=1;i<=r;++i)
        {
//            printf("%d ",a[i]);
            maxn=maxa(a[i]%k,maxn);
        }
//        printf("\n");
//        printf("1 %d ",maxn);
    }
    else
    {
//        printf("2 ",maxn);
        for(i=x+1;i<=y;++i) maxn=maxa(maxn,ans[i][k]);
        for(i=1;i<=x*B;++i) maxn=maxa(maxn,a[i]%k);
        for(i=y*B;i<=r;++i) maxn=maxa(maxn,a[i]%k);
    }
    printf("%d\n",maxn);
}
return 0;
}

```

```

//折射
#include<cstdio>
#include<cstring>
#include<cctype>
#include<iostream>
#include<bitset>
#include<algorithm>
#define int long long
using namespace std;
const int MAXN = 6060;
const int M = 1000000007;

template<typename T>inline void read(T &x)
{

```

```

    int c=getchar();
    x=0;
    T neg=0;
    while(!isdigit(c)) neg|=(c^'-'),c=getchar();
    while(isdigit(c)) x=(x<<1)+(x<<3)+(c^48),c=getchar();
    if(neg) x=(~x)+1;
}

int n,f[MAXN][2];

struct biao
{
    int x,y;
}a[MAXN];

inline bool cmp(biao x,biao b)
{
    return x.x<b.x;
}

inline int maxa(int x,int y)
{
    return x>y ? x : y;
}

signed main()
{
    freopen("refract.in","r",stdin);
    freopen("refract.out","w",stdout);
    read(n);
    register int i,j;
    for(i=1;i<=n;++i)
        read(a[i].x),read(a[i].y);
    sort(a+1,a+n+1,cmp);
    for(i=1;i<=n;++i)
    {
        f[i][0]=f[i][1]=1;
        for(j=i-1;j-->0)
            a[j].y<a[i].y ? f[i][0]=(f[i][0]+f[j][1])%M : f[j][1]=(f[i][0]+f[j]
[1])%M;
    }
    int ans=M-n;
    for(i=1;i<=n;++i) ans=(ans+f[i][0]+f[i][1])%M;
    printf("%lld\n",ans);
    return 0;
}

```



