

2022.11.14考试总结

以为自己摆烂了，没想到成绩有一点出乎意料。

T1.最短母串(substr)

看到字符串就难受。先看到题的时候有两种想法，一种是用KMP去做，另外一种和最长相同子链相似。前一种想的是每次都去进行KMP操作，再构造一个新串，再与下一个串进行KMP操作，但是问题是构造这个新串，于是我转战最长相同子链，想的是找到两个串的最长相同子链，然后根据这个去构造一个符合要求的串，但是又因为要考虑字典序，又卡在了构造上。

最后只能骗分了，于是我特判了 $n=1$ 和 $n=2$ 的情况， $n=1$ 时输出这个串本身， $n=2$ 时输出两个串的最长相同子链，其他情况就找最长的链并输出。骗了20分。

最后看题解发现，这道题是AC自动机，又重新复习了一下（约等于重新学习）

T2.要养花(flower)

这道题一眼看到很开心，因为能暴力打。后面看到是区间操作，以为可以线段树，结果发现是要取模的，线段树如果要这样实现很困难，要是写了还不如暴力。于是最终还是打了暴力，还是非常纯正的纯暴力，稍微卡了卡看起来并没有什么大用的常，一测拿了20分，后面似乎改数据了，然后同样的码测出40分。

考后发现要分块做，以前只打过一道分块的题，印象不深刻，又去找了Siri问，感觉还是不太清楚，半懂不懂，又问了孙巨。最后还是调一调改对了。就是先预处理，把整个分成若干个块，对每个块内处理出不同模数时的最大值，后面再在要求的区间进行寻找，要分类讨论，稍微要推一下。写了这个题还是更了解了分块，还有就是打暴力真的很有必要。

T3.折射(refract)

这道题好坑.....读题就读了好久，还没太懂，试图推样例发现自己一直找不完所有情况，试图递推，最开始按x从小到大排了个序去写，结果统计的时候发现错了，又换成按y从小到大排，结果还是错的，有点崩溃。干脆直接想个办法骗分，果不其然爆零了。

下午改题就先改了这道题，是dp！去问了好多人，一遍又一遍地听，感觉他们都快被问崩溃了，不过幸好最后还是懂了。定义了一个 $dp[i][0/1]$ 表示从左边和从右边转过来的个数，将每个点按照x从小到大进行排序，在倒序枚举这个地方听了好多遍，还有前缀和优化，可以将dp的效率从 (n^3) 变为 (n^2) 。因为可以知道光线是不断左右转移的，很容易推出了dp方程，写出来也特别短。

T4.画作(paint)

这道题是考试最大的意外收获。最开始打了一个bfs，找到有多少个四联通的1的块就输出多少，但是没办法对付样例，于是在bfs里又加了一个条件，处理上下左右四个方向还要找左上左下右上右下四个方向，但是这四个方向与上下左右是分开的，如果走了这四个方向在每次统计的时候还要再单独再加一。居然一测得了56分，是真的没想到啊。隔壁管巨打的dfs好像只得了42分，看来这道题bfs确实好像要强一点。

总结：时间分配比较合理，暴力和骗分基本都得了分，除了T3，不过策略大体上是对的，宽搜也没有打错，还是不错的，但是这次的dp还是没写出来，不过还好最后改对理解了。这次考试真的让我知道，暴力和搜索真的很重要。

最后挂一个表格：

	T1	T2	T3	T4	总分
预估得分	10	20~40	0	未知	40~60
实际得分	20	20	0	56	96

挂出T1,T2,T3的AC代码：

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<queue>
using namespace std;
template<typename T>inline void read(T &x){
    x=0;
    T neg=0;
    int c=getchar();
    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,x,l;
int to,ct,tot;
int dz[610][30],fa[(1<<12|1)*610],state[610],fail[610],ans[(1<<12|1)*610];
bool b[610][1<<12|1];
char anss[610],s[55];
queue<int> q,q1,q2;
void failure(){
    for(int i=0;i<26;i++){
        if(dz[0][i]) q.push(dz[0][i]);
    }
    while(!q.empty()){
        int x=q.front();
        q.pop();
        for(int i=0;i<26;i++){
            if(dz[x][i]){
                fail[dz[x][i]]=dz[fail[x]][i];
                state[dz[x][i]]|=state[dz[fail[x]][i]];
                q.push(dz[x][i]);
            }
            else dz[x][i]=dz[fail[x]][i];
        }
    }
}
int main(){
    freopen("substr.in","r",stdin);
    freopen("substr.out","w",stdout);
    read(n);
    for(int i=1;i<=n;i++){
        scanf("%s",s);
        l=strlen(s);
        int x=0;
        for(int j=0;j<l;j++){
            if(!dz[x][s[j]-'A']) dz[x][s[j]-'A']=++ct;
            x=dz[x][s[j]-'A'];
        }
        state[x]|=(1<<(i-1));
    }
}
```

```

    }
    failure();
    q1.push(0),q2.push(0);
    b[0][0]=1;
    int t=0;
    while(!q1.empty()){
        int x=q1.front(),y=q2.front();
        q1.pop(),q2.pop();
        if(y==(1<<n)-1){
            while(t){
                anss[++to]=ans[t],t=fa[t];
            }
            for(int i=to;i;i--) putchar(anss[i]+'A');
            return 0;
        }
        for(int i=0;i<26;i++){
            if(!b[dz[x][i]][y|state[dz[x][i]])]{
                b[dz[x][i]][y|state[dz[x][i]])=1;
                q1.push(dz[x][i]);
                q2.push(y|state[dz[x][i]]);
                fa[++tot]=t,ans[tot]=i;
            }
        }
        t++;
    }
    return 0;
}

```

```

#include<iostream>
#include<cstdio>
#include<cstring>
#define b 1000
#define maxn 100010
using namespace std;
template<typename T>inline void read(T &x){
    x=0;
    T neg=0;
    int c=getchar();
    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,maxx,a[maxn];
int ans[110][maxn],maxa[maxn];
inline int mmax(int x,int y){
    return x>y?x:y;
}
int main(){
    freopen("flower.in","r",stdin);
    freopen("flower.out","w",stdout);
    register int i,j,k;
    int nn=maxn-10;
    read(n),read(m);
    for(i=1;i<=n;++i)
        read(a[i]);
    int blo=(n-1)/b+1;
    for(i=0;i<blo;++i){

```

```

        memset(maxa,0,sizeof(maxa));
        for(j=i*b+1;j<=n&& j<=(i+1)*b;++j)
            maxa[a[j]]=a[j];
        for(j=1;j<=nn;++j)
            maxa[j]=mmax(maxa[j],maxa[j-1]);
        for(j=1;j<=nn;++j)
            for(k=0;k<=nn;k+=j)
                ans[i][j]=mmax(ans[i][j],maxa[min(j+k-1,nn)]-k);
    }
    while(m--){
        static int l,r,k;
        read(l),read(r),read(k);
        int x=(l/b)+1,y=(r/b);
        maxx=0;
        if(x==y+1)
            for(i=l;i<=r;++i) maxx=mmax(maxx,a[i]%k);
        else{
            for(i=x;i<y;++i) maxx=mmax(maxx,ans[i][k]);
            for(i=1;i<=x*b;++i) maxx=mmax(maxx,a[i]%k);
            for(i=r;i>=y*b;--i) maxx=mmax(maxx,a[i]%k);
        }
        printf("%d\n",maxx);
    }
    return 0;
}

```

```

#include<iostream>
#include<cstdio>
#include<algorithm>
#define maxn 6010
#define mo 1000000007
using namespace std;
template<typename T>inline void read(T &x){
    x=0;
    T neg=0;
    int c=getchar();
    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;
long long dp[maxn][2],ans;
struct zb{
    int x,y;
}a[maxn];
inline bool cmp(zb aa,zb bb){
    return aa.x<bb.x;
}
int main(){
    freopen("refract.in","r",stdin);
    freopen("refract.out","w",stdout);
    read(n);
    for(int i=1;i<=n;i++){
        read(a[i].x),read(a[i].y);
    }
    sort(a+1,a+n+1,cmp);
    for(int i=1;i<=n;i++){

```

```
dp[i][0]=dp[i][1]=1;
for(int j=i-1;j>=1;j--){
    if(a[i].y>a[j].y) dp[i][0]=(dp[i][0]+dp[j][1])%mo;
    else dp[j][1]=(dp[j][1]+dp[i][0])%mo;
}
}
ans=mo-n;
for(int i=1;i<=n;i++){
    ans=(ans+dp[i][0]+dp[i][1])%mo;
}
printf("%11d\n",ans);
return 0;
}
```

既然大家都要贴图，那也贴吧



