



rubish 的博客

test20230222

posted on 2023-02-23 22:00:53 | under 未分类 (.#type=未分类) | 编辑 (<https://www.luogu.com.cn/blogAdmin/article/edit/547501>) | 0

这次考试比较简单，所以分数比较高

$T1$

分析：贪心一波，鹰只会向上飞或向右飞或斜上飞，正确性十分显然；

所以每一行只会走一次斜线

又一个贪心，走的斜线越多，路程越短，同样显然

所以，来想一想，把每一行看做一个状态，对每个行观察是否能取一条斜线，找出斜线最多的路径，是不是有点像 LIS

于是你写出了如下代码

记得注意这题的输入格式，起点是 (n, m) ，终点是 $(1, 1)$ ，十分奇怪，我在这里看了很久

源代码 - C++

```

n=read();m=read();K=read();
for(int i=1,u,v;i<=K;i++){
    u=read();v=read();
    a[v]=-u;
}
int len=1;
h[1]=a[m];
for(int i=m;i>=1;i--){
    int k=lower_bound(h+1,h+1+len,a[i]);
    if(a[i]<h[k]){
        h[k]=a[i];
    }
    if(k==len+1){
        len++;
        h[len]=a[i];
    }
}
}

```

(这串代码是我手胡地，编译都过不了，明白我意思就行)

恭喜，你想出了一个错误但是接近正解的想法

ps:我在这里卡了很久，直到手膜出一组样例才发现—— — — — — — — — — —

同一行可以有多个斜线！！！！！！！！！！

于是问题又来了，我们该怎样取舍这些斜线呢？

等等，也许不需要取舍，我们在求lis时，当 $a[i] < a[j]$ 时就进行更改，把他改一下

源代码 - C++

```

if(a[j].x<a[i].x&&a[j].y<a[i].y)
    f[i]=max(f[i],f[j]+1);

```

然后为了确保读入顺序，我们需要小排个序

源代码 - C++

```

inline cmp(node a,node b){
    if(a.x==b.x)return a.y>b.y;//tmdzhelikenlewo lianggexiaoshi
    return a.x<b.x;
}

```

注意是 $a.y > b.y$ ，本体读入有坑，又卡了10min

源代码 - C++

```
#include<bits/stdc++.h>

using namespace std;

#define int long long

inline int read(){
    int a=0;int f=1;char ch=getchar();
    while(ch<'0' || ch>'9') {if(ch=='-') f=-1;ch=getchar();}
    while(ch>='0'&&ch<='9') {a=(a<<1)+(a<<3)+ch-'0';ch=getchar();}
    return a*f;
}

struct node{
    int x,y;
}a[5005];

double eps=1.4142135;
int K,n,m,h[1500500];
inline cmp(node a,node b){
    if(a.x==b.x)return a.y>b.y;//tmdzhelikenlewojianggexiaoshi
    return a.x<b.x;
}

signed main(){
    freopen("fly.in","r",stdin);
    freopen("fly.out","w",stdout);
    n=read();m=read();
    K=read();
    for(int i=1;i<=K;i++)f[i]=1;
    for(int i=1;i<=K;i++){
        a[i].x=read();
        a[i].y=read();
    }
    sort(a+1,a+1+K,cmp);
    int len;
    for(int i = 1; i <= K; i ++ ) {
        int pos = lower_bound(h + 1, h + 1 + len, a[i].y) - h;
        h[pos] = a[i].y;
        len = max(len, pos);
    }
    //cout<<len<<endl;
    double s=(n+m)*100-len*200+len*eps*100;
    int t=s*10;
    if(t%10>=5)t+=10;
    cout<<t/10;
    return 0;
}
```

T2

看到 $n \leq 5$,直接狂喜, dfs安排上

题不难, 但是有点小坑,

比如

源代码 - C++

```
if(a[id].m[i-x+1][j-y+1]==0)continue;
```

等于0的时候直接跳过就行, 不需要进行清空

然后开始直接开始搜索就好了。一个剪枝都不用加, 搜到了就输出然后`exit(0)` 就好了。

源代码 - C++

```
#include<bits/stdc++.h>

using namespace std;

//#define int long long

inline int read(){
    int a=0;int f=1;char ch=getchar();
    while(ch<'0' || ch>'9') {if(ch=='-') f=-1;ch=getchar();}
    while(ch>='0'&&ch<='9') {a=(a<<1)+(a<<3)+ch-'0';ch=getchar();}
    return a*f;
}

int n,vis[5005],sl;
struct node{
    int m[20][20];
    int chang,kuan;
}a[3200];
int g[505][50];
bool check(){
    for(int i=1;i<=sl;i++){
        for(int j=1;j<=sl;j++){
            if(g[i][j]==0)return 0;
        }
    }
    return 1;
}

void print(){
    for(int i=1;i<=sl;i++){
        for(int j=1;j<=sl;j++){
            printf("%d",g[i][j]);
        }
        printf("\n");
    }
}

bool kefang(int x,int y,int id){
    for(int i=x;i<x+a[id].chang;i++){
        for(int j=y;j<y+a[id].kuan;j++){
            if(a[id].m[i-x+1][j-y+1]==0)continue;
            if(i>sl || j>sl)return 0;
            if(g[i][j]!=0)return 0;
        }
    }
    return 1;
}

void filld(int x,int y,int id){
    for(int i=x;i<x+a[id].chang;i++){
        for(int j=y;j<y+a[id].kuan;j++){
            if(a[id].m[i-x+1][j-y+1]==0)continue;
```

```
        if(i>s1||j>s1)continue;
        g[i][j]=id;
    }
}
}
void clear(int x,int y,int id){
    for(int i=x;i<x+a[id].chang;i++){
        for(int j=y;j<y+a[id].kuan;j++){
            if(a[id].m[i-x+1][j-y+1]==0)continue;
            if(i>s1||j>s1)continue;
            g[i][j]=0;
        }
    }
}
void dfs(int step){
    if(step==n){
        print();
        exit(0);
    }
    if(step>n){
        printf("-1");
        return;
    }
    for(int i=1;i<=s1;i++){
        for(int j=1;j<=s1;j++){
            if(g[i][j]==0){
                for(int k=1;k<=n;k++){
                    //cout<<n<<endl;
                    //if(i==1&&j==3)cout<<k<<" hfdjskfh"<<endl;
                    if(vis[k])continue;
                    if(kefang(i,j,k)){
                        vis[k]=1;
                        filld(i,j,k);
                        //print();
                        dfs(step+1);
                        clear(i,j,k);
                        vis[k]=0;
                        //cout<<"hfdjskhfdjks"<<endl;
                    }
                }
            }
        }
    }
}
signed main(){
    freopen("puzzling.in","r",stdin);
    freopen("puzzling.out","w",stdout);
    n=read();
```

```

for(int i=1,l,r;i<=n;i++){
    l=read();r=read();
    for(int j=1;j<=l;j++){
        for(int k=1;k<=r;k++){
            scanf("%1d",&a[i].m[j][k]);
            if(a[i].m[j][k])sl++;
        }
    }
    a[i].chang=l;a[i].kuan=r;
}
//cout<<sl<<endl;
sl=(int)sqrt(sl);
//cout<<sl<<endl;
dfs(0);
printf("-1");
return 0;
}

```

T3

分析：

显然，当 a 的对手是 b 时， b 的对手是 a

又显然，当自己轮空后，之后的就是一样的

手膜几组数据，可以得出这样一个公式

第 t 天第 i 队的对手为

$$a[(t - b[i] + b[m] + n - 1) \bmod n + 1]$$

源代码 - C++

```

#include<bits/stdc++.h>

using namespace std;

#define int long long

inline int read(){
    int a=0;int f=1;char ch=getchar();
    while(ch<'0' || ch>'9') {if(ch=='-') f=-1;ch=getchar();}
    while(ch>='0' && ch<='9') {a=(a<<1)+(a<<3)+ch-'0';ch=getchar();}
    return a*f;
}

int n,m,t,a[500500],b[50050];
signed main(){
    freopen("match.in","r",stdin);
    freopen("match.out","w",stdout);
    n=read();m=read();t=read();
    for(int i=1;i<=n;i++)a[i]=read();
    int l=0;
    for(int i=1;i<=n;i++){
        if(a[i]==m){
            l=i;
            break;
        }
    }
    for(int i=1;i<=l;i++){
        a[n+i]=a[i];
    }
    for(int i=1;i<=n;i++){
        if(t>=i)b[a[i]]=a[l+t-i];
        else b[a[i]]=a[l+t+n-i];
    }
    for(int i=1;i<=n;i++){
        printf("%lld ",b[i]);
    }
    return 0;
}

```

$T4$

事实上，这是一道结论题

$$A = \max(a_i + a_{i-1}, \frac{sum}{\lfloor \frac{n}{2} \rfloor})$$

具体的证法点这里 (<https://www.luogu.com.cn/problem/solution/P4409>)

源代码 - C++

```
#include<stdio.h>
#include<math.h>
int ans;
int a[20001],m,n,i,j;
int main()
{
    scanf("%d",&n);
    for (i=1;i<=n;++i) scanf("%d",&a[i]);
    for (i=1;i<n;++i)
        if (a[i]+a[i+1]>ans) ans=a[i]+a[i+1];
    for (i=1;i<=n;++i) m+=a[i];
    double mm=m,nn=n,tem;
    tem=ceil(mm/((int)(nn/2)));
    if ((int)tem>ans) ans=(int)tem;
    printf("%d",ans);
    return 0;
}
```

有什么感想？

发射评论！



在洛谷，
享受 Coding 的欢乐

2013-2020，洛谷 (<https://www.luogu.com.cn>) © Developed by the Luogu Dev Team (<https://github.com/luogu-dev>). Site Map ([_sitemap](#))

Blog theme 'Luogu3' By @kkksc03