



rubish 的博客

test20230219

posted on 2023-02-19 22:16:38 | under 未分类 (.#type=未分类) | 编辑 (<https://www.luogu.com.cn/blogAdmin/article/edit/546079>) | 0

$T1$

题目 (<https://www.luogu.com.cn/problem/P1583>)

分析：送分题，将题意模拟一下，就出来了

要注意 $sort$ 排序的 cmp 第一关键字为 w ,从大到小，第二关键字为 num ，从小到大

源代码 - C++

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

using namespace std;

#define int long long

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,k,e[20];
struct node{
    int w,d,num,c;
}p[500500];
bool cmp2(node l,node r){
    if(l.w==r.w)return l.num<r.num;
    return l.w>r.w;
}

signed main(){
    freopen("photo.in","r",stdin);
    freopen("photo.out","w",stdout);
    n=read();k=read();
    for(int i=1;i<=10;i++)e[i]=read();
    for(int i=1;i<=n;i++){
        p[i].w=read();
        p[i].num=i;
    }
    sort(p+1,p+1+n,cmp2);
    for(int i=1;i<=n;i++){
        p[i].d=i;
        p[i].c=((p[i].d-1)%10)+1;
        p[i].w+=e[p[i].c];
    }
    sort(p+1,p+1+n,cmp2);
    for(int i=1;i<=k;i++){
        printf("%lld ",p[i].num);
    }
    return 0;
}
```

T_2

考场上调了两个小时才调出来的我是个跪物

题目 (<https://www.luogu.com.cn/problem/P1875>)

题意：给出一对关系 $a, b - c$,形如

$$dis[c] = \min(dis[a] + dis[b], dis[c])$$

的式子，可以化为最短路问题；

由于数据范围小于1000邻接矩阵是个不错的选择

读入

源代码 - C++

```
for(int i=0;i<n;i++){
    w[i]=read();
    ans[i]=1;
}
int a,b,c;
memset(dis,-1,sizeof dis);
while(cin>>a>>b>>c){
    dis[a][b]=c;
    dis[b][a]=c;
}
```

在每一次收敛中如果 w 的值变小 ans 的值重置为联通两端之积，如果收敛的值等于原值

, ans 加上联通两边之只因，

源代码 - C++

```
w[dis[j][k]]=w[j]+w[k];
w[dis[k][j]]=w[j]+w[k];
ans[dis[j][k]]=ans[k]*ans[j];
ans[dis[k][j]]=ans[k]*ans[j];
```

源代码 - C++

```
ans[dis[j][k]]+=ans[k]*ans[j];
```

具体操作类似 $spfa$ +弗洛伊德

源代码 - C++

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

using namespace std;

#define int long long
#define inf 0x3f3f3f3f

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,w[500500],ans[500500],vis[500500];
vector<pair<int,int> >vec[50500];
int dis[1205][1205];

signed main(){
    freopen("syrup.in","r",stdin);
    freopen("syrup.out","w",stdout);
    n=read();
    for(int i=0;i<n;i++){
        w[i]=read();
        ans[i]=1;
    }
    int a,b,c;
    memset(dis,-1,sizeof dis);
    while(cin>>a>>b>>c){
        dis[a][b]=c;
        dis[b][a]=c;
    }
    int maxl;
    for(int i=0;i<n;i++){
        maxl=inf;
        int k;
        for(int j=0;j<n;j++){
            if(vis[j])continue;
            if(w[j]<maxl){
                k=j;
                //cout<<k<<" "<<i<<" "<<maxl<<endl;
                maxl=w[j];
                //cout<<maxl<<endl;
            }
        }
        vis[k]=1;
        for(int j=0;j<n;j++){
            if(vis[j]&&dis[j][k]!=-1){
```

```
        if(w[dis[j][k]]==w[j]+w[k]){
            ans[dis[j][k]]+=ans[k]*ans[j];
            //ans[dis[k][j]]+=ans[k]*ans[j];
            //int ll=dis[j][k];
            //cout<<w[ll]<<" "<<ans[ll]<<endl;
        }
        if(w[dis[j][k]]>w[j]+w[k]){
            w[dis[j][k]]=w[j]+w[k];
            w[dis[k][j]]=w[j]+w[k];
            ans[dis[j][k]]=ans[k]*ans[j];
            ans[dis[k][j]]=ans[k]*ans[j];
            //int ll=dis[j][k];
            //cout<<w[ll]<<" "<<ans[ll]<<endl;
        }
    }
}
cout<<w[0]<<" "<<ans[0]<<endl;
return 0;
}
```

但是考场上本来第一思路是 dfs ，可是写挂了，考后就重写一遍、

源代码 - C++

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

using namespace std;

#define int long long
#define inf 0x3f3f3f3f

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,w[500500],ans[500500],f[500500];
vector<pair<int,int> >vec[500500];
int dfs(int u){
    if(f[u]<inf){
        return f[u];
    }
    f[u]=w[u];
    for(auto i:vec[u]){
        int l=i.first,r=i.second;
        int t=dfs(l)+dfs(r);
        if (t < f[u]) {
            f[u] = t;
            ans[u]=ans[l]*ans[r];
        }
        else if(t==f[u]){
            ans[u]+=(ans[l]*ans[r]);
        }
    }
}

signed main(){
    freopen("syrup.in","r",stdin);
    freopen("syrup.out","w",stdout);
    n=read();
    for(int i=0;i<n;i++){
        w[i]=read();
        ans[i]=1;
    }
    memset(f,0x3f,sizeof f);
    int a,b,c;
    while(cin>>a>>b>>c){
        vec[c].push_back({a,b});
    }
    dfs(0);
    cout<<f[0]<<' '<<ans[0]<<endl;
```

```
    return 0;  
}
```

T3

大意：给你一堆一样的树枝，从中截取下 N 段做魔杖，每一段有一个魔法值和长度，要求一共可以截取下多少符合条件切树枝两两不完全重合的树枝使魔法值最大

DP问题

令 $f[i][j]$ 为起点小于 i 终点小于 j 的最大值

先将前面的状态传递过来

$$f[i][j] = \max(f[i-1][j], f[i][j-1]);$$

如果长度满足条件，因为可以重合，只要不完全包含就行，所以转移方程里是 $f[i-1][j-1]$

$$f[i][j] = \max(f[i][j], f[i-1][j-1] + \text{summ}[j] - \text{summ}[i-1]);$$

源代码 - C++

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

using namespace std;

#define int long long

inline int read(){
    int x=0,f=1;char ch=getchar();
    while (ch<'0' || ch>'9'){if(ch=='-') f=-1;ch=getchar();}
    while (ch>='0'&&ch<='9'){x=x*10+ch-48;ch=getchar();}
    return x*f;
}

int n,low,hi,l[1500],m[1500],suml[1500],summ[1500],f[1300][1300];
signed main(){
    n=read();low=read();hi=read();
    for(int i=1;i<=n;i++){
        l[i]=read();
        suml[i]=suml[i-1]+l[i];
    }
    for(int i=1;i<=n;i++){
        m[i]=read();
        summ[i]=summ[i-1]+m[i];
    }
    for(int i=1;i<=n;i++){
        for(int j=i;j<=n;j++){
            int len=suml[j]-suml[i-1];
            f[i][j]=max(f[i-1][j],f[i][j-1]);
            if(len>=low&&len<=hi){
                f[i][j]=max(f[i][j],f[i-1][j-1]+summ[j]-summ[i-1]);
            }
        }
    }
    cout<<f[n][n]<<"\n";
    return 0;
}
```

T4

题目 (<https://www.luogu.com.cn/problem/P1585>)

暴搜非常好写，轻松可以骗80分，只用两个数组存下每一步的坐标就行了

源代码 - C++


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

using namespace std;

#define int long long

inline int read(){
    int x=0,f=1;char ch=getchar();
    while (ch<'0' || ch>'9'){if(ch=='-') f=-1;ch=getchar();}
    while (ch>='0' && ch<='9'){x=x*10+ch-48;ch=getchar();}
    return x*f;
}

int n,m,k1,k2,mod,ans=0x3f3f3f3f,vis[500][500],px[2550],py[2550];
int dx[]={1,-1,0,0},dy[]={0,0,1,-1};
inline int calc(int x1,int y1,int x2,int y2){
    return k1*abs(x1-x2)+k2*abs(y1-y2);
}

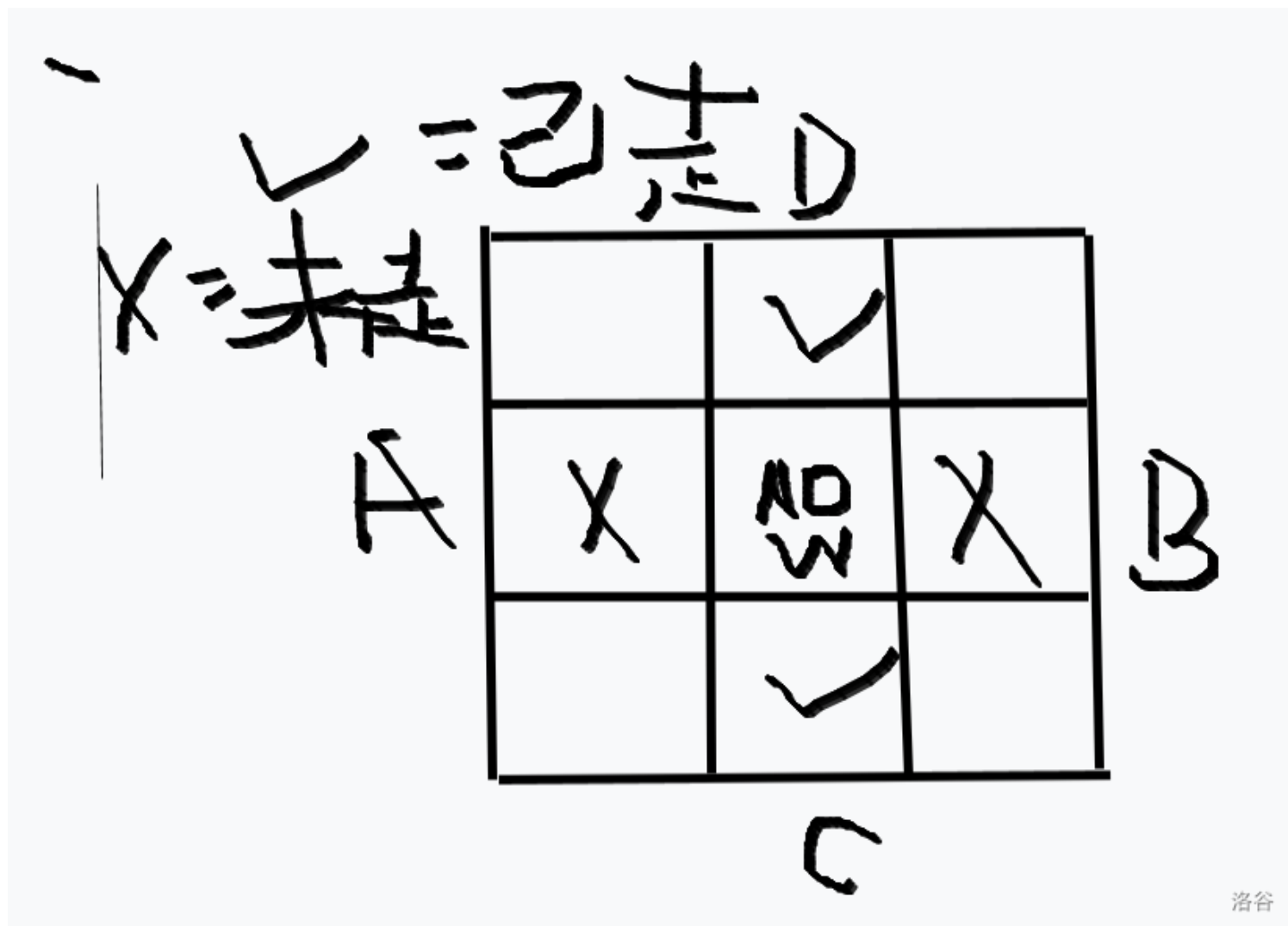
inline void dfs(int x,int y,int step,int sum){
    int res=0;
    if(step<=mod){
        px[step]=x;
        py[step]=y;
    }
    else res=calc(x,y,px[step-mod],py[step-mod]);
    if(sum>=ans) return ;
    if(step==n*m) {
        ans=min(ans,max(sum,res));
        return;
    }
    for(int i=0;i<=3;i++){
        int tx=x+dx[i],ty=y+dy[i];
        if(tx<=0 || ty<=0 || tx>n || ty>m)continue;
        if(vis[tx][ty])continue;
        vis[tx][ty]=1;
        dfs(tx,ty,step+1,max(sum,res));
        vis[tx][ty]=0;
    }
}

signed main(){
    n=read();m=read();k1=read();k2=read();
    mod=n*m/2;
    init();
    dfs(1,1,1,0);
    cout<<ans;
    return 0;
}
```

但是一百分的剪枝有点思维难度，首先注意这句话

每个格子必须且仅能到过1次

当我们走到一个位置时,如果已经走过它的上下,却没有走过它左右的任意一格,或者相反,已经走过它的左右,却没有走过它上下的任意一格,那么一定无法走完魔法阵,直接 return;



证明：如图，从A走到B的路径隔绝了C和D，因此不可能在不经过A到B路径的情况下同时走到C和D，

源代码 - C++

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

using namespace std;

#define int long long

inline int read(){
    int x=0,f=1;char ch=getchar();
    while (ch<'0' || ch>'9'){if(ch=='-') f=-1;ch=getchar();}
    while (ch>='0' && ch<='9'){x=x*10+ch-48;ch=getchar();}
    return x*f;
}

int n,m,k1,k2,mod,ans=0x3f3f3f3f,vis[500][500],px[2550],py[2550];
int dx[]={1,-1,0,0},dy[]={0,0,1,-1};
inline int calc(int x1,int y1,int x2,int y2){
    return k1*abs(x1-x2)+k2*abs(y1-y2);
}

inline void init() {
    memset(vis,0x3f,sizeof(vis));
    for(int i=1;i<=n;i++) {
        for(int j= 1;j<=m;j++) {
            vis[i][j]=0;
        }
    }
    vis[1][1] = 1;
}

inline void dfs(int x,int y,int step,int sum){
    if (vis[x+1][y] && vis[x-1][y] && !vis[x][y+1] && !vis[x][y-1]) return ;
    if (!vis[x+1][y] && !vis[x-1][y] && vis[x][y+1] && vis[x][y-1]) return ;

    int res=0;
    if(step<=mod){
        px[step]=x;
        py[step]=y;
    }
    else res=calc(x,y,px[step-mod],py[step-mod]);
    if(sum>=ans) return ;
    if(step==n*m) {
        ans=min(ans,max(sum,res));
        return;
    }
    for(int i=0;i<=3;i++){
        int tx=x+dx[i],ty=y+dy[i];
        if(tx<=0 || ty<=0 || tx>n || ty>m)continue;
        if(vis[tx][ty])continue;
        vis[tx][ty]=1;
        dfs(tx,ty,step+1,max(sum,res));
        vis[tx][ty]=0;
    }
}
```

```
    }  
}  
signed main(){  
    n=read();m=read();k1=read();k2=read();  
    mod=n*m/2;  
    init();  
    dfs(1,1,1,0);  
    cout<<ans;  
    return 0;  
}
```

有什么感想？

发射评论！



(<https://www.luogu.com.cn/space/show?uid=378346>)

small_rubbish (<https://www.luogu.com.cn/space/show?uid=378346>) 2023-02-20 08:20:57

Orz



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) (<https://github.com/luogu-dev>). [Site Map](#) ([_sitemap](#))

Blog theme 'Luogu3' By @kkksc03