



最大子矩阵问题



西南大学附属中学
High School Affiliated to Southwest University

a_{11}	a_{12}	a_{13}	...	a_{1n}
a_{21}	a_{22}	a_{23}	...	a_{2n}
a_{31}	a_{32}	a_{33}	...	a_{3n}
a_{41}	a_{42}	a_{43}	...	a_{4n}
...
a_{n1}	a_{n2}	a_{n3}	...	a_{nn}



枚举连续的行列

然后用一个维度来表示

$b[k]$:第 k 列的和

$a_{31}+a_{41}$

$a_{32}+a_{42}$

$a_{33}+a_{43}$

...

$a_{3n}+a_{4n}$

西 | 大 | 附 | 中 | 信 | 息 | 学 | 竞 | 赛 |
High School Affiliated to Southwest University



参考代码



西南大学附属中学
High School Affiliated to Southwest University

```
#include<bits/stdc++.h>
using namespace std;
int a[55][55];
int b[55];
int f[55]; //最大子段和的状态
int main(){
    memset(b,0,sizeof(b));
    int n,m;
    cin>>n>>m;
    for(int i=0;i<n;i++){
        for(int j=0;j<m;j++){
            cin>>a[i][j];
        }
    }
    int ant=-100;
    for(int i=0;i<n;i++){
        memset(b, 0, sizeof(b));
        for(int j=i;j<n;j++){ //i到n行的值相加, 求最大值
            for(int k=0;k<m;k++){
                b[k]+=a[j][k];
            }
            int ans=-0x3f;
            f[0]=b[0];
            for(int l=0;l<m;l++){ //求最大子段和
                if(f[l-1]>=0) f[l]=b[l]+f[l-1];
                else f[l]=b[l];
                ans=max(ans,f[l]);
            }
            ant=max(ans,ant);
        }
    }
    cout<<ant;
    return 0;
}
```



尼克的任务



西南大学附属中学
High School Affiliated to Southwest University

题意：有n个进行的任务，如果某时刻如果有任务必须完成，有多个任务在同一时刻开始则选一个进行，没有的话就休息。求最大的休息时间。

状态定义 $dp[i]$ ：第i时刻的最大休息时间

这道题画图会发现，如果正向选择任务，当前任务选或不选其实会对后面的任务产生影响

后效性

怎么办呢？

正难则反

逆序推，
三种情况：

- 1、第i分钟如果有任务，那么任务是一定要做的 $dp[i] = \max(dp[i], dp[i+t[j]])$ (i时刻有任务)
- 2、如果有多个任务，那就比较每个任务做完之后的最大空闲
- 3、第i分钟没有任务，那么空闲就为前从i+1开始的最大空闲再加上1分钟的空闲。 $dp[i] = dp[i+1] + 1$



参考代码



西南大学附属中学
High School Affiliated to Southwest University

```
#include<bits/stdc++.h>
using namespace std;
int a[10001][10001];
int cnt[10001],f[10001];
int main(){
    int n,k;
    scanf("%d%d",&n,&k);
    for(int i=1,start,end;i<=k;i++){
        scanf("%d%d",&start,&end);
        a[start][++cnt[start]]=end;
    }
    for(int i=n;i>=1;i--){
        if(!cnt[i])f[i]=f[i+1]+1; //没有任务
        else{ //有任务
            for(int j=1;j<=cnt[i];j++){
                f[i]=max(f[i],f[i+a[i][j]]);
            }
        }
    }
    printf("%d",f[1]);
    return 0;
}
```

西 | 大 | 附 | 中 | 信 | 息 | 学 | 竞 | 赛 |
High School Affiliated to Southwest University

Thanks

For Your Watching

