Single Problem Set (1)

1.方格取数

- **Date:** 2020/5/12
- **Tag:** DP
- **Source:** https://www.luogu.com.cn/problem/P1004
- **Problem:** 设有 N × N 的方格图 (N≤9)，我们将其中的某些方格中填入正整数，而其他的方格中则放入数字 0。某人从图的左上角的 A 点出发，可以向下行走，也可以向右走，直到到达右下角的 B 点。在走过的路上，他可以取走方格中的数（取走后的方格中将变为数字00）。此人从 A 点到 B 点共走两次，试找出 2 条这样的路径，使得取得的数之和为最大。
- **Solution:** 不妨两次同时走，则两次总步数是相同的，只需要用 ans[step][x1][x2] 存储 “ 步数为step，第一、二次行动的横坐标分别为 x1，x2 ”时的最优解，递归即可。

```
#include<stdio.h>
#include<string.h>
#define condition(con,tr,fa) ((con)?(tr):(fa))
#define max(a,b) condition(a>b,a,b)
#define min(a,b) condition(a<b,a,b)
#define reset(arr) memset(arr,0,sizeof(arr))

int map[10][10];
int ans[20][10][10];
int ansflag[20][10][10];
int n;

int ansf(int step,int x1,int x2){
    if(step==0)return map[0][0];
    if(ansflag[step][x1][x2]>0){
        return ans[step][x1][x2];
    }
    int y1=step-x1;
    int y2=step-x2;

    int temp=0;
    if(x1>0 && x2>0) temp=max(temp,ansf(step-1,x1-1,x2-1));
    if(x1>0 && y2>0) temp=max(temp,ansf(step-1,x1-1,x2));
    if(y1>0 && x2>0) temp=max(temp,ansf(step-1,x1,x2-1));
    if(y1>0 && y2>0) temp=max(temp,ansf(step-1,x1,x2));
    ans[step][x1][x2]=temp+map[x1][y1]+map[x2][y2]*condition(x1==x2,0,1);
    ansflag[step][x1][x2]=1;

    return ans[step][x1][x2];
}
```

```
int main(){
    reset(map);
    reset(ans);
    reset(ansflag);

    scanf("%d",&n);

    int x,y,val=1;
    while(x!=0 || y!=0 || val!=0){
        scanf("%d%d%d",&x,&y,&val);
        map[x-1][y-1]=val;
    }
    /*
    printf("\n");
    for(int i=0;i<n;i++){
        for(int j=0;j<n;j++){
            printf("%d ",map[i][j]);
        }
        printf("\n");
    }printf("\n");
    */

    printf("%d",ansf(n*2-2,n-1,n-1));
}
```

2.传纸条

- **Date:** 2020/5/14
- **Tag:** DP
- **Source:** https://www.luogu.com.cn/problem/P1006
- **Problem:** 设有 M × N 的方格图 (M, N≤50)□所有方格填入非负整数；其中，保证左上右下为0。从左上到右下选择两条路径，每条必须满足要么向右要么向下，且两条不能有交叉。求路径数字之和最大值。
- **Solution:** 解类似1，不过在递归时，取消了 x1-1==x2-1 一项。另外，需要注意的是，多维数组不要忘了哪个分量为行或者列。