

2020 年 7 月 30 日

1.P1157 组合的输出

算法思路：

这道题数据结构与算法的实验课做过~BUT当时是用的algorithm库里的函数next_permutation(课上百度的嘤嘤嘤)。今天会用dfs做出来拉！！

主函数里面从1 ~ n开始dfs，数组a用来存储当前的组合结果，cnt是计数当前的组合已经有多少位数，到达r位数就输出结果，回溯。

代码：

```
#include <iostream>

using namespace std;

int n,r;
int a[25];
void dfs(int val,int cnt){

    a[cnt] = val;

    if(cnt >= r){
        for(int i = 1; i <= r; i++){
            cout<<" ";
            if( a[i] < 10)  cout << " " << a[i];
            else cout << a[i];
        }

        cout << endl;
        return;
    }

    for(int i = val+1;i <= n;i++ ){
        dfs(i,++cnt);
        cnt--;
    }
    return;
}

int main(){

    cin >> n >> r;

    for(int i = 1; (i+r-1) <= n; i++){ // 从i开始的组合输出r位之后，超过n就退出
```

```
        dfs(i,1);  
    }  
  
    return 0;  
}
```

Accepted截图：



备注：

1. 这道题输出格式好坑啊~输出是两位数的话就间隔一个空格，否则两个

2.P3392 涂国旗

算法思路：

看了一些题解才大概明白咋做，不看不会嘤嘤嘤

$w[i][0]$ 代表 i 行全涂成白色的块数， $w[i][1]$ 代表 i 行全涂成蓝色的块数， $w[i][2]$ 代表 i 行全涂成红色的块数。 $f[i][0]$ 代表累计到当前 i 行涂成白色的块数，因为前一个状态只能是涂白色，所以 $f[i][0] = f[i-1][0] + w[i][0]$ 。 $f[i][1]$ 代表累计到当前 i 行涂成蓝色的块数，因为前一个状态可以是白色或者蓝色，所以 $f[i][1] = \min(f[i-1][0], f[i-1][1]) + w[i][1]$ ，即前一个状态的最小值 + 这一行是涂蓝色的块数。 $f[i][2]$ 代表累计到当前 i 行涂成红色的块数，因为前一个状态可以是红色或者蓝色，所以 $f[i][2] = \min(f[i-1][1], f[i-1][2]) + w[i][2]$ ，即前一个状态的最小值 + 这一行是涂红色的块数。

因为第一行和最后一行一定是白色和红色，所以先初始化第一行，即 $f[0][0]$ ，然后是最后一行，选择倒数第二行是涂蓝色还是红色的最小块数，再加上最后一行涂成红色的块数即是结果。

代码：

```

#include <iostream>
#include <vector>
using namespace std;

#define N 55
#define INF 1e9
int w[N][3], f[N][3];

int main(){
    int n,m,res = INF;
    cin >> n >> m;

    vector< vector<int> > flag(n);

    for(int i = 0; i < n; i++){
        string temp;
        cin >> temp;
        for(int j = 0; j < m; j++){
            if(temp[j] == 'W') w[i][0]++; //这一行的白色个数
            if(temp[j] == 'B') w[i][1]++; //这一行的蓝色个数
            if(temp[j] == 'R') w[i][2]++; //这一行的红色个数
        }

        w[i][0] = m - w[i][0]; //这一行涂成白色需要涂的块数
        w[i][1] = m - w[i][1]; //这一行涂成蓝色需要涂的块数
        w[i][2] = m - w[i][2]; //这一行涂成红色需要涂的块数
    }

    //dp
    f[0][0] = w[0][0]; //初始化，当前行涂白需要涂的块数
    f[0][1] = f[0][2] = INF; //初始化，第一行只能涂白
    for(int i = 1; i < n-1; i++){
        f[i][0] = f[i-1][0] + w[i][0]; //累计到当前行涂白
        需要涂的块数
        f[i][1] = min( f[i-1][0] , f[i-1][1] ) + w[i][1]; //累计到当前行涂蓝
        需要涂的块数
        f[i][2] = min( f[i-1][1] , f[i-1][2] ) + w[i][2]; //累计到当前行涂红
        需要涂的块数
    }

    for(int i = 1; i <= 2; i++){
        res = min( res , f[n-2][i] ); //倒数第二行涂蓝或涂红
    }

    cout << res + w[n-1][2]; //最后一行一定涂红

    return 0;
}

```

Accepted截图：

wongsiyoung

所属题目

P3392 涂国旗

评测状态

Accepted

评测分数

100

提交时间

2020-07-30 13:52:49