

愉快的代码【8.3】

【7.27】我们特意加入了题解系统，也就是说，姐姐也会重新做一遍给你们的题目（包括选做），然后在第二天的题目前给出姐姐自己的代码和注释作为题解或参考

如果觉得自己的代码略为臃肿，可以参考对比一下姐姐的代码；

如果觉得姐姐的代码不如自己的优秀，也可以尽情地嘲讽姐姐～

【7.30】我们特意加入了团队系统，因为感觉到你们有点像是独立学习的样子，比如说姐姐和你们之间有交流，但是你们之间有没有交流呢姐姐就感受不到啦

所以正好在洛谷上发现了一个团队系统，我们可以在这上面布置作业呀（当然姐姐也会继续以 pdf 形式布置作业，你们也还是要以 pdf 形式交作业哈），然后你们就可以在上面看到其它妹妹们的代码呀（包括 AC 代码和还未 AC 的整个过程的代码和分数呀），觉得她们表现不够自己好的话，就可以在群里尽情地嘲笑她们呀～

然后那上面还有一个比赛功能哇，具体形式和我们平时的机考差不多，暑假差不多结束了我们也会有一次期末模拟机考的哈～

如果你们开心的话，你们也可以联合起来给姐姐布置一次平时的作业呀，或者给姐姐安排一次机考呀，你们都是团队的管理员了哈

【8.1】准备给你们留个有趣的团队大作业：给姐姐安排一次机考～

具体时间、题数、难度、知识点待定～

【8.3】经过了某些人性与道德的思考，得出了一个奇怪的想法：

“我今天把代码解决了，明天姐姐的代码还有兴趣看嘛”

那就当天放出来好啦～

同样地：如果觉得姐姐的代码不如自己的优秀，也可以尽情地嘲讽姐姐～

8.2问题1：

```
/*
    洛谷P2404：自然数的拆分问题
    思想：DFS。由于要优先输出字典序小的序列，因此不适合用BFS。
    时间复杂度：O(2^n)
*/

#include <iostream>
#include <algorithm>
#include <string>
using namespace std;

void dfs(int n,int m,string s) { // 剩余数字、拆分出的数字、拆分形成的字符串
    s = s+" "+(char)(m+48);
```

```

    if (n == 0 and s.length() != 4) { // s.length() != 4: 排除本身情况
        cout << s.substr(3) << endl; // 由于写路径在前，因此要删掉字符串前端的一些多余的东西
        return;
    }
    for (int i=m; i<=n; i++) dfs(n-i,i,s);
}

int main() {
    int i,j,n;
    cin >> n;
    dfs(n,1,"");
}

```

8.2问题2:

```

/*
    洛谷P3817: 小A的糖果
    思想: 贪心。从左到右遍历糖果盒，然后超出的话优先吃掉右边盒子的糖果。理论上除了第一个盒子之外，只吃右边盒子的糖果就可以解决超出的问题，思考一下叭
    时间复杂度: O(n)
*/

#include <iostream>
#include <algorithm>
#define ll long long
using namespace std;

int main() {
    ll i,j,n,x,ans = 0;
    cin >> n >> x;
    ll a[n];
    for (i=0; i<n; i++) cin >> a[i];
    for (i=1; i<n; i++) { // 这里我没有看第一个盒子有没有超出范围，实际上应该要看的
        if (a[i]+a[i-1] > x) {
            ans += a[i]+a[i-1]-x;
            a[i] -= a[i]+a[i-1]-x; // 吃右边的
        }
    }
    cout << ans << endl;
}

```

今天的题目:

知识点: 回溯

应思蓉妹妹的要求，这两天都是关于回溯的内容

回溯是一种配合 DFS 的技术（或者说是思想）

在深搜的时候可能会改变某些状态

比如说搜索到了某个位置，把它的访问数组 `visit[i] = 1`;

那么回溯的思想为：在一步深搜结束后，把访问数组 `visit[i]` 重置

因此来确保这个方向的深搜过程搜索之后改变的某些状态在结束这个方向的深搜之后重置为原来的样子

即不影响下个方向的深搜

回溯有两大经典：迷宫问题 and 八皇后问题

因此这两天就是这些问题的研究探讨啦

今天先是迷宫问题：

1、<https://www.luogu.com.cn/problem/P1605>

2、<https://www.luogu.com.cn/problem/P1238>

今天的答案：

8.3问题1：

```
/*
    洛谷P1605：迷宫
    思想：回溯入门题，其实没什么思想，乖乖配合深搜即可～
    时间复杂度： $O(3^{(n*m)})$ ，指数
*/

#include <iostream>
#include <algorithm>
#define ll long long
using namespace std;

int t,n,m,x,y,z,w,ans;
int dx[4] = {1,-1,0,0},dy[4] = {0,0,1,-1}; // 四个搜索方向
int a[6][6];
void dfs(int x,int y) {
    if (x == z and y == w) { // 终点
        ans++;
        return;
    }
    for (int i=0; i<4; i++) {
        if (x+dx[i] > 0 and x+dx[i] <= n and y+dy[i] > 0 and y+dy[i] <= m and
a[x+dx[i]][y+dy[i]] == 0) {
            a[x+dx[i]][y+dy[i]] = 1; // 标注经过这一点
            dfs(x+dx[i],y+dy[i]); // 深搜
            a[x+dx[i]][y+dy[i]] = 0; // 回溯
        }
    }
}
```

```

}

int main() {
    int i,p,q;
    cin >> n >> m >> t;
    cin >> x >> y >> z >> w; // 起点、重点
    for (i=0; i<t; i++) {
        cin >> p >> q;
        a[p][q] = 1; // 障碍当作已经过
    }
    a[x][y] = 1; // 起点经过啦
    dfs(x,y);
    cout << ans << endl;
}

```

8.2问题2:

```

/*
    洛谷P1238：走迷宫
    思想：上一题的升级版，按顺序输出路径。回溯部分基本和上一题一样
    时间复杂度：O(3^(n*m))，指数
*/

#include <iostream>
#include <algorithm>
#define ll long long
using namespace std;

int t,n,m,x,y,z,w,ans;
int dx[4] = {0,-1,0,1},dy[4] = {-1,0,1,0}; // 优先顺序：左上右下
int a[15][15];
void dfs(int x,int y,string s) {
    s = s+"->"+(char)(x)+","+(char)(y)+";"; // string当数组
    if (x == z and y == w) { // 终点
        for (int i=2; i<s.length(); i++) { // 由于写路径在前，因此要删掉字符串前端的一些多余的东西
            if (s[i] > 0 and s[i] < 15) cout << (int)s[i]; // 输出坐标
            else cout << s[i]; // 输出符号
        }
        cout << endl;
        ans++; // 有路
        return;
    }
    for (int i=0; i<4; i++) {
        if (x+dx[i] > 0 and x+dx[i] <= n and y+dy[i] > 0 and y+dy[i] <= m and a[x+dx[i]][y+dy[i]] == 1) {
            a[x+dx[i]][y+dy[i]] = 0; // 这道题0代表经过
            dfs(x+dx[i],y+dy[i],s);
        }
    }
}

```

```

        a[x+dx[i]][y+dy[i]] = 1; // 一样的回溯
    }
}

int main() {
    int i,j;
    cin >> n >> m;
    for (i=1; i<=n; i++)
        for (j=1; j<=m; j++)
            cin >> a[i][j];
    cin >> x >> y >> z >> w;
    a[x][y] = 0; // 起点经过啦
    dfs(x,y,"");
    if (ans == 0) cout << -1 << endl; // 没路输出-1
}

```

Interesting thing:

<https://www.luogu.com.cn/contest/32578>