二、程序阅读题（共 35 分）

```
#include <iostream>
using namespace std;
void inputAndBuild();
void turn(int i, int j);
void flip(int s);
int Win();
void dfs(int s, int b);
int c,m,**map;
int main()
{
    cin >> m;
     c = 2 * m * m + 1;
     map = new int* [m+1];
     for (int i = 0; i <= m; i++)
          map[i] = new int[m + 1];
      inputAndBuild();
      dfs(0, 0);
      if (c == m * m * 2 + 1)
          cout << "Impossible" << endl;
      else
          cout << c << endl;
       return 0;
}
void inputAndBuild()
{
    char c;
    int i, j;
    for (i = 0; i < m; i++){
        for (j = 0; j < m; j++){
            cin >> c;
            if (c == 'w')
                map[i][j] = 0;
            else
                map[i][j] = 1;
          }
       }
}
void turn(int i, int j)
{
    if (i >= 0 && i <= m - 1 && j >= 0 && j <= m - 1)
        map[i][j]= !map[i][j];
     return;
}
void flip(int s)
{
```

```
46.         int i = s / m;
47.         int j = s % m;
48.         turn(i, j);
49.         turn(i + 1, j);
50.         turn(i, j + 1);
51.         turn(i - 1, j);
52.         turn(i, j - 1);
53.         return;
54. }
55. int Win()
56. {
57.         int i, j, s1 = 0;
58.         for (i = 0; i < m; i++)
59.               for (j = 0; j < m; j++)
60.                     s1 += map[i][j];
61.          if (s1 % (m * m))
62.                return 0;
63.          else
64.                return 1;
65. }
66. void dfs(int s, int b)
67. {
68.          if (Win()){
69.                if (c > b)
70.                      c = b;
71.                return;
72.          }
73.          if (s >= m * m)
74.                return;
75.          dfs(s + 1, b);
76.          flip(s);
77.          dfs(s + 1, b + 1);
78.          flip(s);
79.          return;
80. }
```

（1）设程序的输入如下，请写出程序执行到 17 行时变量 map 的值。（5 分）

4

bwwb

bbwb

bwwb

bbbb

答案：

Map=1001
1101
1001
1111

（2）函数 dfs 的参数 s 的含义：__map 中单元格的索引，从 0 到 m*m-1__；dfs 的参数 b 的含义：_调用 flip 对 map 中元素取反的次数_；函数 Win 在什么情况下会返回 1：__map 中的元素为全 0 或全 1 _；函数 flip 的作用：__将参数 s 对应的 map 的单元格的元素，以及该元素的上、下、左、右的元素进行取反__；程序输出值 c 的含义：_将 map 从初始状态按照 flip 函数的规则进行取反，变成全 1 或全 0 所需的最少次数___；（每空 2 分，共 10 分）。

（3）分析函数 dfs 的时间复杂度，写出分析过程。（10 分）

答：$T(0)=2T(1)+O(m^2)=2*2T(2)+2*m^2+m^2=\ldots=O(m^2 2^{m*m-1})$ 或 $O(m^2 2^{m*m})$

或：

$T(s) = 2*T(s+1)+ O(m^2)$　$O(m^2)$是每轮中 Win()的时间复杂性

$T(m*m)=O(1)$ 或 $O(0)$　递归出口的时间复杂性

或：

画出递归树进行分析。

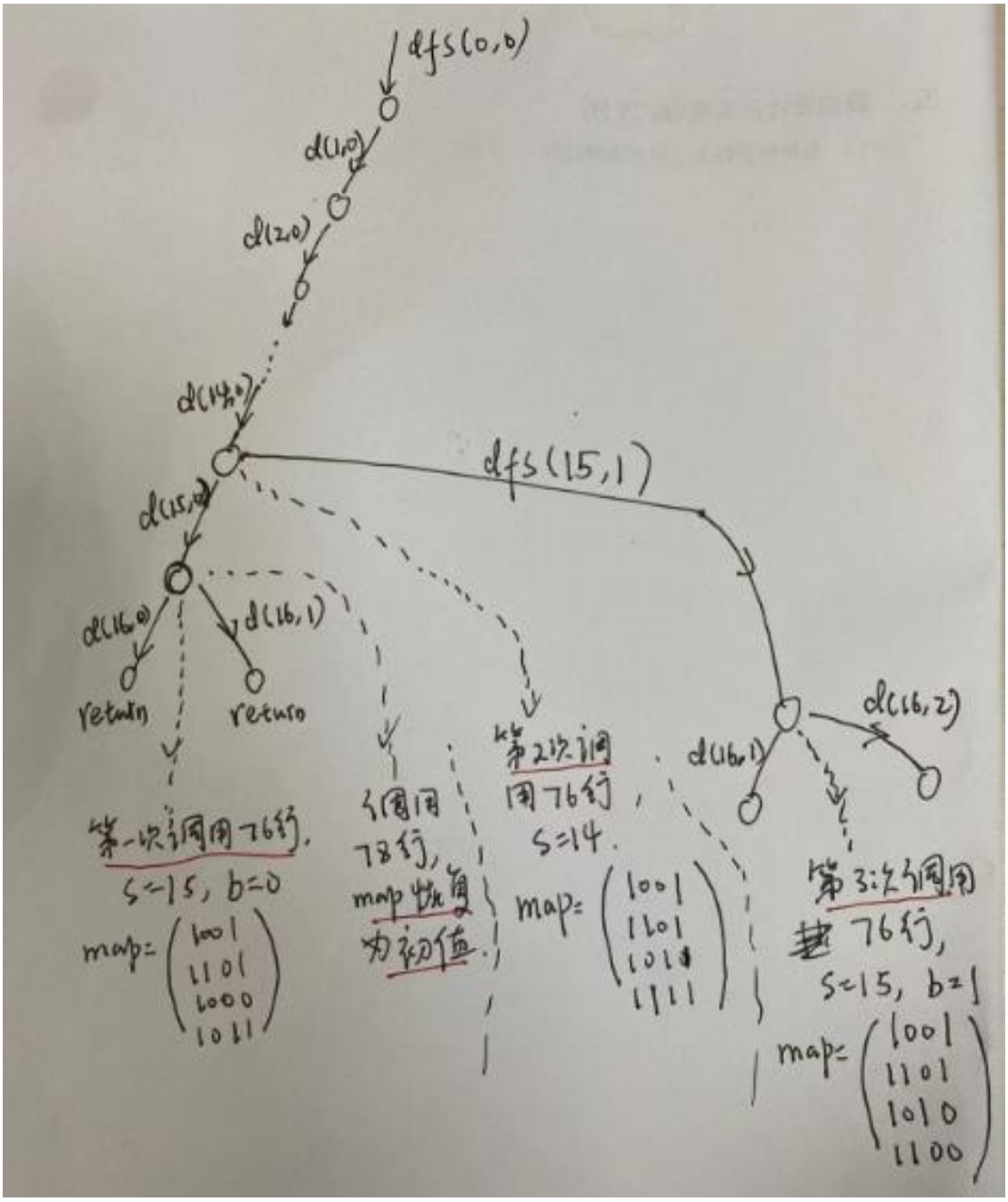
（4）采用（1）中的输入，请写出程序第 3 次执行 76 行代码后变量 s（3 分）、b（3 分）和 map（4 分）的值。

答：s=15，b=1，

Map= 1001
1101
1010
1011

注意：下面的图是按照以前的输入绘制的。以前的输入是：

$$map = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$



三、算法分析及实现题（共 45 分）

某公司进行组合投资，把钱同时投入到多个项目，并希望获取最大的收益。在已知该公司希望投资的最大金额数和投资的项目数的情况下，请编写程序计算其所能获得的最大收益。

输入：

输入的第一行包含两个整数 n 和 m，分别表示该公司的投资资金数和希望投资的项目数。其后的 n 行，每行有 m 个整数，分别表示将金额 1、2、…、n 单独投资到 m 个项目时的收益（投资为 0 时 m 个项目的收益均为 0）。每次组合投资必须将 n 用完，组合投资中某个项目的投资额可以为 0。

输出：

输出一行，表示该公司进行组合投资所能获得的最大收益。

样例输入：

3 2

3 2

3 5

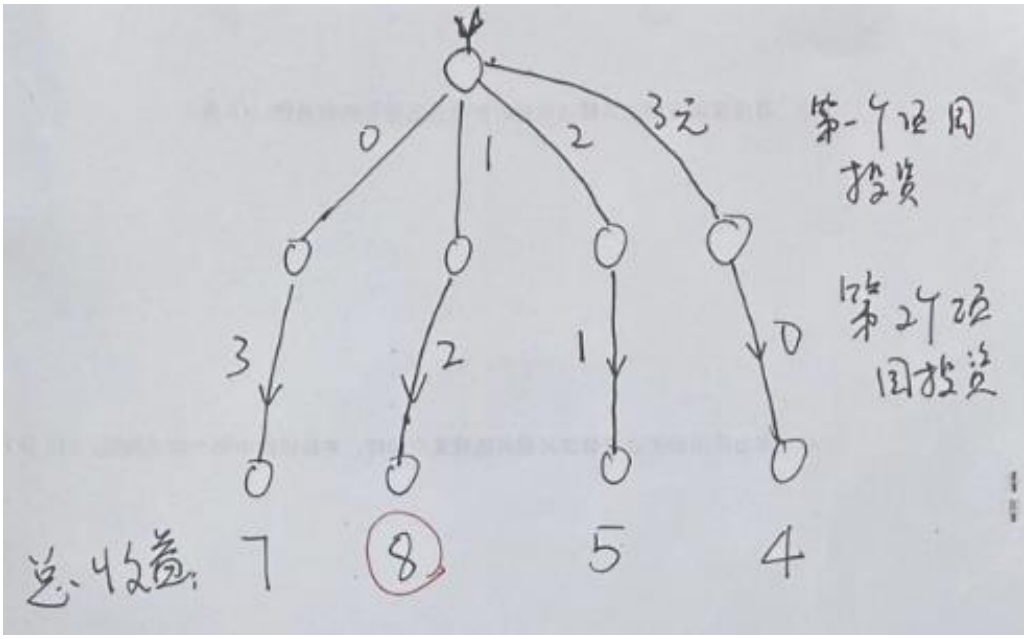
4 7

样例输出：

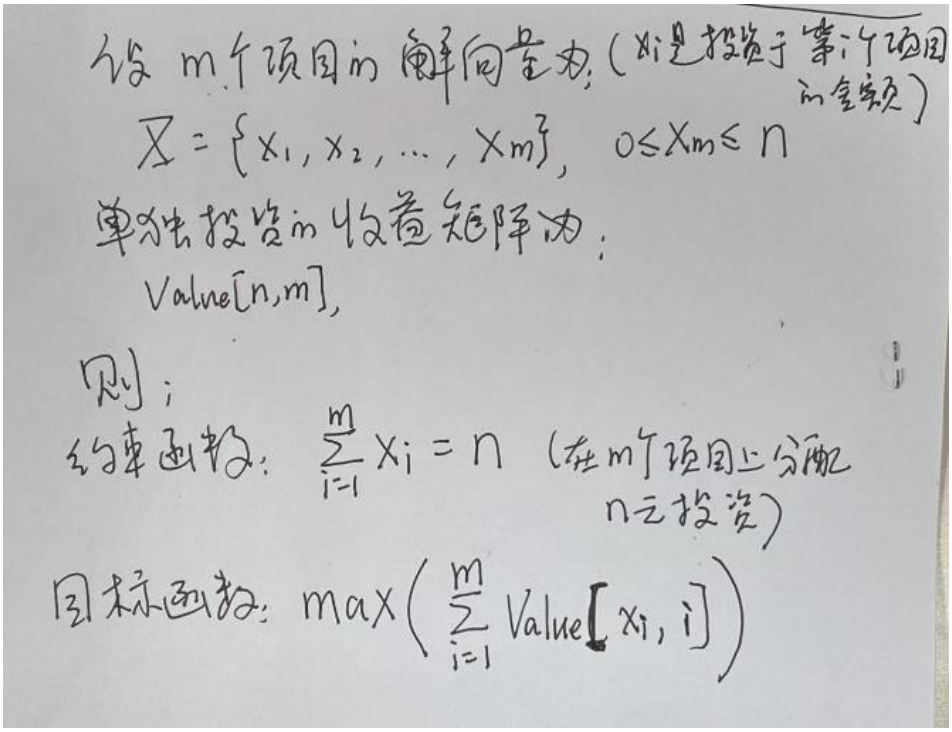
8

要求：（共 45 分）

（1）采用回溯法分析上面的测试用例，画出搜索空间树。（10 分）



（2）写出采用回溯法求解该问题时的约束条件和目标函数。（5 分）



（3）采用回溯法编写程序实现上述题目要求。（15 分）

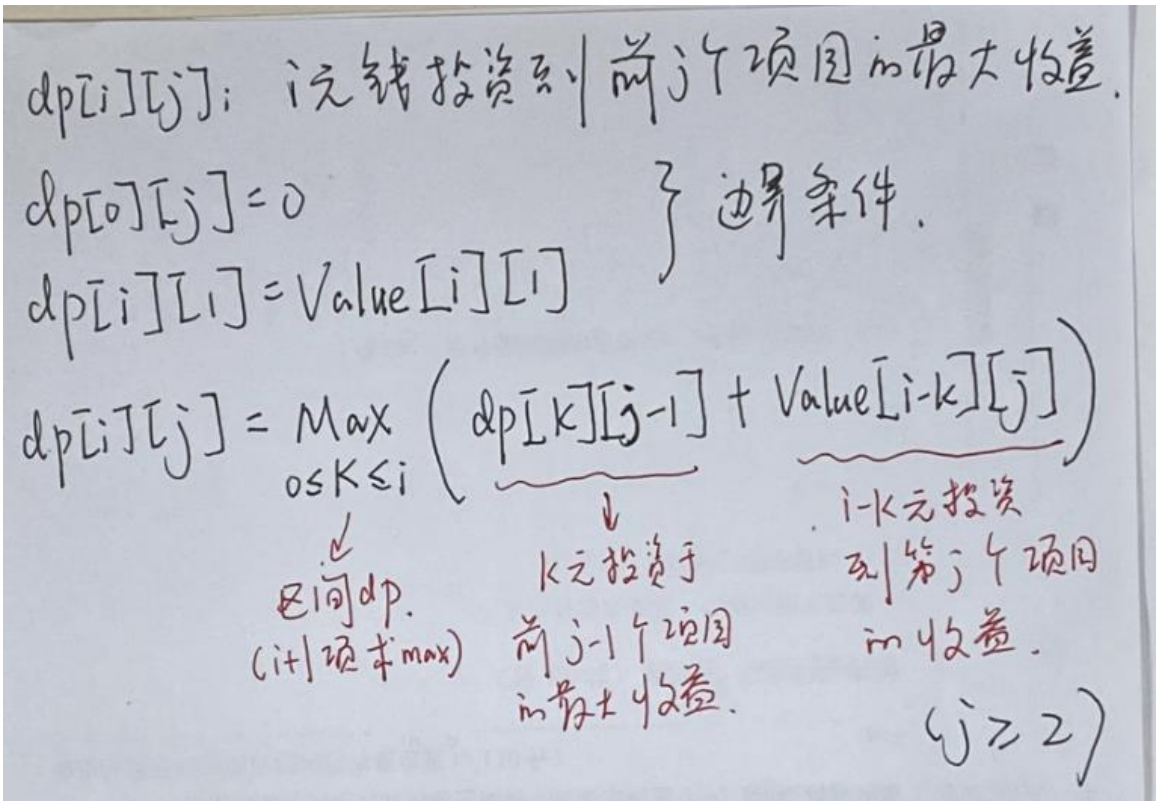
参考代码：

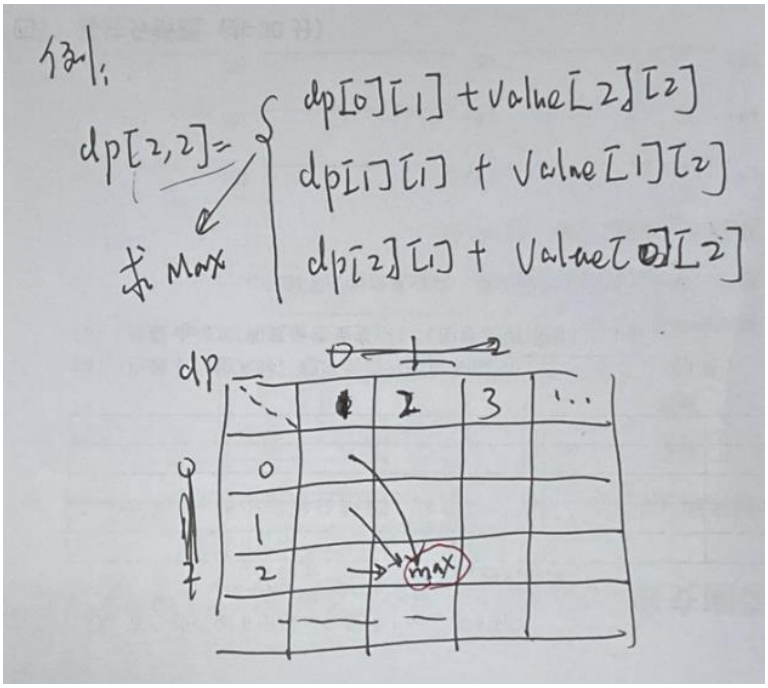
```
void inputAndBuild();
void dfs(int level, int curr_invest, int curr_profit);
int n, m; //n:投资总额; m: 项目数;
int max_profit; //max_profit: 组合投资最大收益
int profit[50][50];
int main()
{
    cin >> n >> m;
    max_profit = 0;
    inputAndBuild();
    dfs(0, 0, 0);
    cout << max_profit << endl;
    return 0;
}
void inputAndBuild()
{
    int c;
    int i, j;
    for (j = 0; j < m; j++) {
        profit[0][j]=0;   //每个项目投资 0 元时收益为 0
    }
    for (i = 1; i <= n; i++) {
        for (j = 0; j < m; j++) {
            cin >> c;
            profit[i][j] = c;
        }
    }
}

//level: 递归深度（即项目号，范围：0到m-1）；curr_invest：当前总投资；curr_profit：当前总收益
void dfs(int level, int curr_invest, int curr_profit)
{
    int i;
    if (level==m) {
        if ((curr_invest=n) && (curr_profit>max_profit)){
            max_profit=curr_profit;
        }
        return;
    }
    for (i=0 ; i<=n ; i++){   //横向遍历每个项目投资 0 到 n 元的情况
        if ((curr_invest+i)<=n)  //约束函数
        {
            dfs(level+1, curr_invest+i, curr_profit+profit[i][level]);
        }
```

```
    }
}
```

（4）写出采用动态规划求解该问题的递推方程和边界条件（简要描述 dp 数组、数组下标和递推方程的含义）。（10 分）





（5）采用动态规划法编写程序实现上述题目要求。（5 分）（注意：如果未正确完成小题（4），则小题（5）不得分）

略