

# 搜索 - 多源BFS

## 一、AcWing 173. 矩阵距离

### 【题目描述】

给定一个  $N$  行  $M$  列的 01 矩阵  $A$ ， $A[i][j]$  与  $A[k][l]$  之间的曼哈顿距离定义为：

$$\text{dist}(A[i][j], A[k][l]) = |i - k| + |j - l|$$

输出一个  $N$  行  $M$  列的整数矩阵  $B$ ，其中：

$$B[i][j] = \min_{1 \leq x \leq N, 1 \leq y \leq M, A[x][y]=1} \text{dist}(A[i][j], A[x][y])$$

### 【输入格式】

第一行两个整数  $N, M$ 。

接下来一个  $N$  行  $M$  列的 01 矩阵，数字之间没有空格。

### 【输出格式】

一个  $N$  行  $M$  列的矩阵  $B$ ，相邻两个整数之间用一个空格隔开。

### 【数据范围】

$$1 \leq N, M \leq 1000$$

### 【输入样例】

```
1 3 4
2 0001
3 0011
4 0110
```

### 【输出样例】

```
1 3 2 1 0
2 2 1 0 0
3 1 0 0 1
```

### 【分析】

题目要求我们求所有点到最近的为1的点的距离，第一种思路是以每个1为起点跑一遍**BFS**，维护每个为0的点的最小距离，这么做会**TLE**。

我们完全可以认为这道题是多起点问题，也就是说，我们直接将所有为1的点作为起点同时加入到状态队列之中，那么这道题目就解决了。

---

### 【代码】

```
1  #include <iostream>
2  #include <cstring>
3  #include <algorithm>
4  #include <queue>
5  using namespace std;
6
7  typedef pair<int, int> PII;
8  const int N = 1010;
9  char g[N][N];
10 int dis[N][N];
11 int n, m;
12 int dx[4] = { 0, 1, 0, -1 }, dy[4] = { 1, 0, -1, 0 };
13
14 void bfs()
15 {
16     memset(dis, -1, sizeof dis);
17     queue<PII> Q;
18     for (int i = 0; i < n; i++)
19         for (int j = 0; j < m; j++)
20             if (g[i][j] == '1') Q.push({ i, j }), dis[i][j] = 0;
21     while (Q.size())
22     {
23         auto t = Q.front();
24         Q.pop();
25         for (int i = 0; i < 4; i++)
26         {
27             int nx = t.first + dx[i], ny = t.second + dy[i];
28             if (nx >= 0 && nx < n && ny >= 0 && ny < m && !~dis[nx][ny])
29                 Q.push({ nx, ny }), dis[nx][ny] = dis[t.first][t.second]
30 + 1;
31         }
32     }
33
34 int main()
```

```
35 {
36     ios::sync_with_stdio(false);
37     cin >> n >> m;
38     for (int i = 0; i < n; i++) cin >> g[i];
39     bfs();
40     for (int i = 0; i < n; i++)
41     {
42         for (int j = 0; j < m; j++)
43             cout << dis[i][j] << ' ';
44         cout << endl;
45     }
46     return 0;
47 }
```