

动态规划的核心思想

最优子结构：问题的最优解可以由其子问题的最优解构成；

子问题重叠：在递归解决问题时，不同的子问题往往会重复出现。动态规划通过**备忘录**或**状态表**将这些子问题的解缓存下来，避免了重复计算，从而提高效率。

问题 3-7

3-7 汽车加油行驶问题。

问题描述：给定一个 $N \times N$ 的方形网格,设其左上角为起点 \odot ，坐标为 $(1, 1)$ ， X 轴向右为正， Y 轴向下为正，每个方格边长为 1。一辆汽车从起点 \odot 出发驶向右下角终点 \blacktriangle ，其坐标为 (N, N) 。在若干网格交叉点处，设置了油库，可供汽车在行驶途中加油。汽车在行驶过程中应遵守如下规则：

(1) 汽车只能沿网格边行驶，装满油后能行驶 K 条网格边。出发时汽车已装满油,在起点与终点处不设油库。

(2) 当汽车行驶经过一条网格边时，若其 X 坐标或 Y 坐标减小,则应付费用 B ，否则免付费用。

(3) 汽车在行驶过程中遇油库则应加满油并付加油费用 A 。

(4) 在需要时可在网格点处增设油库，并付增设油库费用 C (不含加油费用 A)。

(5) (1) ~ (4) 中的各数 N 、 K 、 A 、 B 、 C 均为正整数。

算法设计：求汽车从起点出发到达终点的一条所付费用最少的行驶路线。

数据输入：由文件 input.txt 提供输入数据。文件的第 1 行是 N 、 K 、 A 、 B 、 C 的值， $2 \leq N \leq 100$ ， $2 \leq K \leq 10$ 。第 2 行起是一个 $N \times N$ 的 0-1 方阵，每行 N 个值，至 $N+1$ 行结束。方阵的第 i 行第 j 列处的值为 1 表示在网格交叉点 (i, j) 处设置了一个油库，为 0 时表示未设油库。各行相邻的 2 个数以空格分隔。

结果输出：将找到的最优行驶路线所需的费用即最小费用输出到文件 output.txt。文件的第 1 行中的数是最小费用值。

输入文件示例

input.txt

9 3 2 3 6

0 0 0 0 1 0 0 0 0

0 0 0 1 0 1 1 0 0

1 0 1 0 0 0 0 1 0

0 0 0 0 0 1 0 0 1

1 0 0 1 0 0 1 0 0

0 1 0 0 0 0 0 1 0

0 0 0 0 1 0 0 0 1

1 0 0 1 0 0 0 1 0

0 1 0 0 0 0 0 0 0

输出文件示例

output.txt

12

备忘录：自顶向下

$f(x, y, 0)$ 表示从(1, 1)到(x, y)所需最少费用， $f(x, y, 1)$ 表示到(x, y)还能行驶的边数， $f(N, N, 0)$ 即为结果。

初始情况： $f(1, 1, 0)=0$, $f(1, 1, 1)=K$

递归式：

$$f(x, y, 0) = \min\{f(x+s[i][0], y+s[i][1], 0)+s[i][2]\} \quad (0 \leq i \leq 3)$$

$$f(x, y, 1) = f(x+s[j][0], y+s[j][1], 1)-1$$

数组s表示行驶方向和行驶费用： $s=\{-1, 0, 0\}, \{0, -1, 0\}, \{1, 0, B\}, \{0, 1, B\}\}$

源代码：dp7.c

时间复杂度： $O(N^4)$

do...while循环最坏情况下需要更新所有状态， $O(N^2)$ ；

每次迭代中，枚举每个格，共 N^2 个，检查4个方向， $O(N^2)$ ；

空间复杂度： $O(N^2)$

三维数组memo，存储每个格的两个状态；二维数组map

问题 3-15

3-15 收集样本问题。

问题描述：机器人 Rob 在一个有 $n \times n$ 个方格的方形区域 F 中收集样本。 (i, j) 方格中样本的价值为 $v(i, j)$ ，如图 3-8 所示。Rob 从方形区域 F 的左上角 A 点出发，向下或向右行走，直到右下角的 B 点，在走过的路上，收集方格中的样本。Rob 从 A 点到 B 点共走 2 次，试找出 Rob 的 2 条行走路径，使其取得的样本总价值最大。

算法设计：给定方形区域 F 中的样本分布，计算 Rob 的 2 条行走路径，使其取得的样本总价值最大。

数据输入：由文件 input.txt 给出输入数据。第 1 行有 1 个正整数 n ，表示方形区域 F 有 $n \times n$ 个方格。接下来每行有 3 个整数，前 2 个数表示方格位置，第 3 个数为该位置样本价值。最后一行是 3 个 0。

结果输出：将计算的最大样本总价值输出到文件 output.txt。

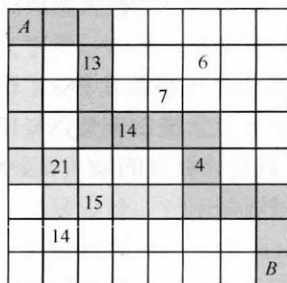


图 3-8 $n \times n$ 个方格的方形区域 F

输入文件示例

input.txt

```
8
2 3 13
2 6 6
3 5 7
4 4 14
5 2 21
5 6 4
6 3 15
7 2 14
0 0 0
```

输出文件示例

output.txt

67

动态规划：自底向上

1. Rob 只能向下/右行走，到 (x, y) 时收集到样本价值只依赖于上一步的选择，该问题具有最优子结构性质；

2. A 到 B 行走 2 次，起点终点固定，可知两条路径的总步数相同，用一个四维矩阵存储采集样本最大价值，设为 $h[x1][y1][x2][y2]$ ，初始状态 $h[1][1][1][1] = v[1][1]$ ；

3. update 通过检查 4 种可能的前一步状态来更新当前状态的最大值

4. $h[n][n][n][n]$ 即为最优解

源代码：dp15.c

时间复杂度： $O(n^4)$

四层嵌套循环

空间复杂度: $O(n^4)$

四维数组 h , 二维数组 v

可能的优化: 将四维状态矩阵 $h[x1][y1][x2][y2]$ 优化为三维矩阵 $h[x1][y1][step]$, 其中 $step$ 表示两条路径总步数, 从而避免冗余存储路径的具体位置。