**1.**

由题意，可以将规则理解为车移动无阻。由于路径长度按走过的格子数计算，可以认为车每次向上下左右四个方向之一移动一个格子。设车初始在左上角，要去往右下角。

**a.**

分析得出：

①走到右下角的路径最短，要求走到其他格子路径也要最短，满足最优子结构

②求到右下角的最短路径数，要反复求解到其他格子的最短路径数，满足重叠子问题

③存在一条贴着棋盘边界的最短路径

④车不会往上或往左这一步

因为一旦做了④，最终在回到右下角的情况下，需要用更多的步数抵消往上或往左的效果，而这显然会比③的路径长，定然不是最短路径。因此：

①车只会向右或向下，换句话说对于某一个车经过的格子(i, j)，车会在某一步时，要么从左边(i, j-1)过来，要么从上面(i-1, j)过来，只有这两种情况。可以说，到达(i, j)的路径数刚好等于到达(i-1, j)的路径数加上(i, j-1)的路径数。

②而到达(0, 0)显然只有一种走法，到第一行或第一列的最短路径显然是从(0, 0)走直线一种走法

③由于起点和终点在棋盘的对角线上，因此到每一个格子的最短路径数呈对角线对称，对角线元素有：



因此，用waycount[i][j]二维数组记录走到(i, j)的最短路径数，有如下关系式：



数组初始化为：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 |  |  |  |  |  |  |  |
| 1 |  |  |  |  |  |  |  |
| 1 |  |  |  |  |  |  |  |
| 1 |  |  |  |  |  |  |  |
| 1 |  |  |  |  |  |  |  |
| 1 |  |  |  |  |  |  |  |
| 1 |  |  |  |  |  |  |  |

由于填写的需要，按从左到右、从上到下填写，遇到对角线元素时取其正上方元素\*2，其余正常填写。填完上三角区域即可。得：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | **2** | 3 | 4 | 5 | 6 | 7 | 8 |
| 1 |  | **6** | 10 | 15 | 21 | 28 | 36 |
| 1 |  |  | **20** | 35 | 56 | 84 | 120 |
| 1 |  |  |  | **70** | 126 | 210 | 330 |
| 1 |  |  |  |  | **252** | 462 | 792 |
| 1 |  |  |  |  |  | **924** | 1716 |
| 1 |  |  |  |  |  |  | 3432 |

因此有3432个最短路径

**b.**

由前面分析可知，车每一步只能向下或向右，使得i+1或j+1，最终让i和j从0到7。因此车在最短路径上移动的特征有：

①总共需要14步

②其中只有7步向下，7步向右

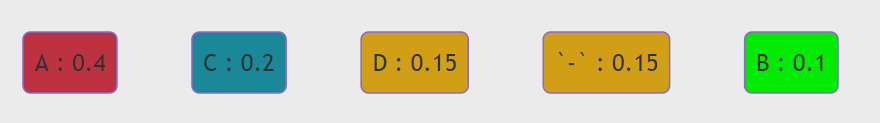
满足这两个条件一定能达成目标，因此求最短路径数时可以对移动的顺序进行排列组合：

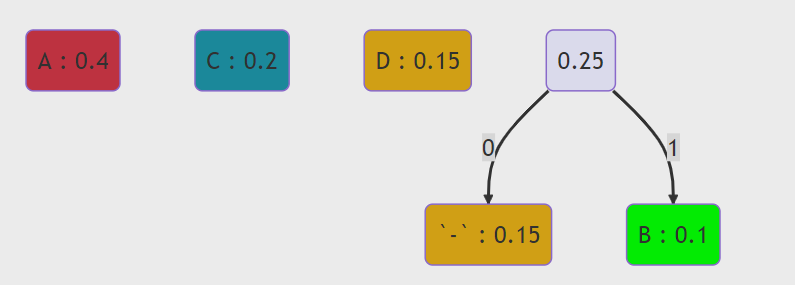


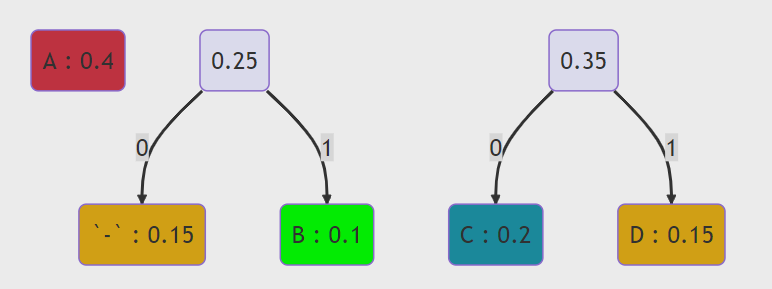
因此有3432个最短路径

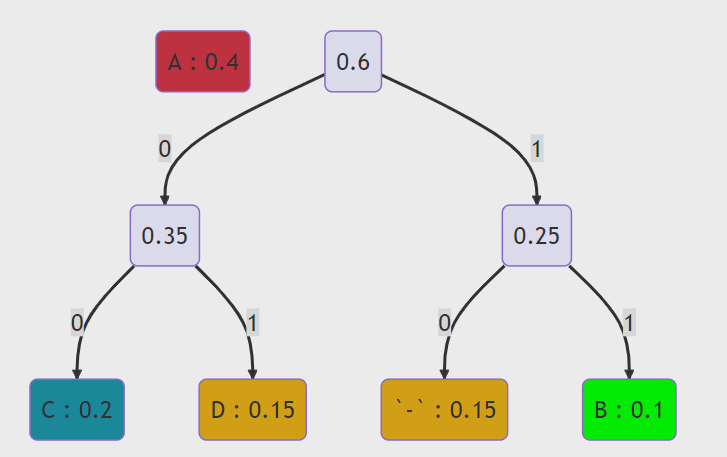
**2.**

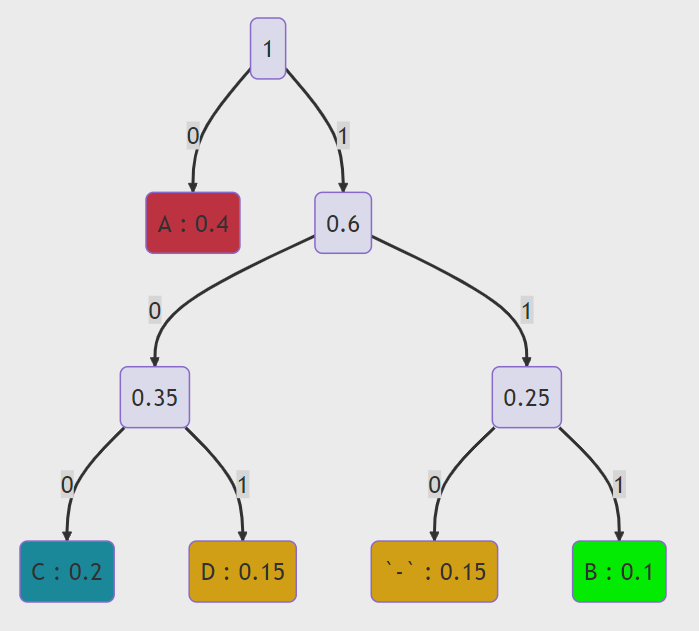
**a.**将各字符出现概率按从大到小排列，构造过程如下：











构造结果：

A：0

B：111

C：100

D：101

-：110

**b.**ABACABAD的编码是：0 111 0 100 0 111 0 101

**c.**编码分成：100 0 101 110 0 101 0，解码为：CAD-ADA