2019 ADA miniHW 3

b07902064 資工二 蔡銘軒

October 4, 2019

(1) From the problem statement, there's no way to reach the cells with obstacle, so we only focus on the cells that can be traveled.
Let (i, j) denote the cell in row i and column j and the count starts from 0. We first deal with the boundary where $i = 0$ or $j = 0$. For obvious reason, there's only one way to reach $(i, 0)$ or $(0, j)$. And for cell (i, j) not on the boundary, we can reach it only from $(i-1, j)$, $(i, j-1)$, $(i-1, j-1)$. Let $dp[i][j]$ be the number of ways to reach (i, j), we have the following formula:

$$dp[i][j] = \begin{cases} 0 & if\ (i,j)\ is\ an\ obstacle \\ 1 & if\ i = 0\ or\ j = 0 \\ dp[i-1][j] + dp[i][j-1] + dp[i-1][j-1] & else \end{cases}$$

(2)

```
DP(n, m):
   Let n be the number of rows, m be the number of columns
   Create an array dp[n][m] and fill it with 0
   for i from 0 to n - 1:
      dp[i][0] = 1
   for j from 0 to m - 1:
      dp[0][j] = 1
   for i from 1 to n - 1:
      for j from 1 to m - 1:
         if (i, j) is not an obstacle:
            dp[i][j] = dp[i - 1][j] + dp[i][j - 1] + dp[i - 1][j -
                1]
   return dp[n - 1][m - 1]
```

reference

none