CCC '12 S5 – Mouse Journey (Editorial)

8 – 15 Marks:

We can use a recursive approach for partial marks (8/15). We must convert the problem statement into an equation. Let $f(r, c)$ be the number of paths for given position (r, c). If (r, c) is ever a position of cat it means there are 0 paths for given (r, c). However otherwise, we can say that the sum of the paths of the position one to the right $(r, c + 1)$ and one to the bottom $(r + 1, c)$ is the number of paths for (r, c). This is because we are only given two paths to choose for given (r, c). From the observations we can build a recurrence of $f(r, c)$.

$$f(r,c) = r(r+1,c) + f(r,c+1)$$

Our recursive structure will look like so:

$$f(r,c) = \begin{Bmatrix} 1 & (r,c) = dest \\ 0 & (r,c) = cat \\ f(r+1,c) + f(r,c+1) & else \end{Bmatrix}$$

However this approach **TLE**'s, memoized or not.

15 – 15 Marks:

To solve for full marks, we must use a dynamic programming approach. We can think a little backwards. Instead of looking forward at possible paths given (r, c), we look precomputed paths which lie at $(r - 1, c)$ (above) and $(r, c - 1)$ (left). Let our state be $dp[i][j]$ which is the optimal path for position (i, j). Also let $dp[1][1] = 1$. Iterating over all values of i an j we can get the value of $dp[i][j]$ from the sum of $dp[i - 1][j]$ and $dp[i][j - 1]$ provided that (i, j) is not a cat.

Time Complexity: $O(RC)$