1. The problem –
   1. Input – A 2D array of grid
   2. Input – A 1D array of pattern
   3. You can only move left, right, up and down
2. The solution
   1. Brute force solution –
      1. Enumerate all subarrays of the 2D grid
      2. Match the pattern with each of these subarrays
      3. Time complexity will be really high.
   2. Faster DP –
      1. We match the first element of the pattern with every entry of 2D grid until we find a match. Then we follow through.
      2. But is it really DP?
      3. DP is made of subproblems and caching
      4. DP is used for solving problems with choices
         1. So technically it is really a choice whether a suffix match an entry in the grid or not
         2. Those choices are the subproblems
3. Time complexity
   1. We iterate through each entry in 2D array = m \* n
   2. In each iteration we match the pattern whose size is |S|
   3. Therefore time complexity = m \* n \* |S|