<https://leetcode.com/problems/minimum-cost-to-cut-a-stick/>

**Approach:** f(i, j)

1. Start with the entire block/ array.
2. Try all possible cuts.
3. Return the minimum cost that is achieved by the particular sequence of cuts .

**Recursive soln:**

1. Make [1 3 6] -> [0 1 3 6 n]; n = size of the stick
2. Take entire block by passing i = 1, j= m-2
3. Try all possible cuts by taking ind = i to j, cost = size (i to j)

**cost = cuts[j+1]-cuts[i-1] + f(i, ind-1) + f(ind+1, j)**

1. Return the min cost

**Tabulation(bottom-up):** i = m-2 to 1 and j = 1 to m-2

1. Initialize dp[m][m] with 0 to counter base case.
2. Use the same recurrence relation to build dp.
3. Return dp[1][m-2]