Problem type 1:

In lab 13, you saw the problem below and the recurrence that computes that problem. In plain English, describe what the call to that recurrence $f(i, j, \ldots)$ means.

(See variants below)

a. **BYD/BYE**

Problem: You are give an integer value x and an array $A[1..n]$ where each element of the array represents a coin denomination:

Recurrence:

$$CC(i,j) = \begin{cases} 1 & \text{if } j = 0 \\ 0 & \text{if } j < 0 \text{ or } (j > 0 \text{ and } i = 0) \\ CC(i, j - A[i]) + CC(i-1, j) & \text{otherwise} \end{cases}$$

In plain english, what does $CC(i, j)$ represent?

Solution: $CC(i, j)$ denotes the number of different ways to make change for j using the first i types of coins($A[1..i]$). ■

b. **BYC/BYF**

Problem: Find maximum value of items that can be fit into knapsack of the defined capacity. You are given: a array of values V where each element corresponds to item i with value $V[i]$, an array of integer weights W where each elements corresponds to item i with weight $W[i]$ and an integer x which corresponds to the capacity of the knapsack.

Recurrence:

$$Sack(i,j) = \begin{cases} 0 & i > n \\ Sack(i+1, j) & W[i] > j \\ \max\{V[i] + Sack(i+1, j - W[i]), Sack(i+1, j)\} & W[i] \le j \end{cases} \quad (1)$$

In plain english, what does $Sack(i, j)$ represent?

Solution: $Sack(i, j)$ is the maximum value of items that can fit in a sack with capacity j, with items $i, \ldots, n$. ■

c. **BYB/BYG**

Problem: You are given a $n \times n$ binary array A and the goal is to find the set of elements within that array that form a square filled with only **1**'s.

Recurrence:

$$LSq(i,j) = \begin{cases} 0 & \text{if } A[i,j] = 0 \quad (2a) \\ A[i,j] & \text{if } i = n \text{ or } j = n \quad (2b) \\ 1 + \min \left\{ \begin{array}{l} LSq(i+1,j) \\ LSq(i,j+1) \\ LSq(i+1,j+1) \end{array} \right\} & \text{otherwise} \quad (2c) \end{cases}$$

In plain english, what does $LSq(i,j)$ represent?

Solution: $LSq(i,j)$ describes the maximum square of **1**'s whose top left corner is at coordinate index $[i,j]$. ■

d. **BYA/BYH**

Problem: You are given an integer x that represents the length of rod you have and an array A where i corresponds to a rod length and $A[i]$ corresponds to the price a rod of that length would fetch. You need to determine the maximum value you can fetch from the rod assuming you cut it optimally.

Recurrence:

$$MV(n) = \max_{0 \le i < n} \big(A[i] + MV(n-i)\big)$$

In plain english, what does $MV(n)$ represent?

Solution: $MV(i)$ denotes the maximum value we can get from a rod of length i. ■