Problem type 1:

Last quiz you were asked to provide the recurrence that describes one of the backtracking problems from Labs 11/12. I will give you the recurrence to that problem below. **Now I want to know the evaluation order of the recurrence.** Specifically I want three things:

- The number of for loops needed to evaluate the recurrence.
- The order of each of those for loops ($1 \to n$, $n \to 1$, $i \to n$, etc.)
- The return value (which value/part of the array do we return)

Not looking for full pseudocode. Just a basic idea of how to memorize the recurrence.

```
A[n][n] = <Base Cases>
for <loop1 conditions>
                              ▷ Fill in if needed
    for <loop2 conditions>
                              ▷ Fill in if needed
        for <loop3 conditions>
                              ▷ Fill in if needed
            Compute
return ##                     ▷ Fill in
```

(See variants below)

a. **BYA & BYH**

Given an array $A[1..n]$ of integers, compute the length of a **longest alternating subsequence**: Let $LAS^+(i,j)$ denote the length of the longest alternating subsequence of $A[i..n]$ whose first element (if any) is larger than $A[j]$ and whose second element (if any) is smaller than its first.

$$LAS^+(i,j) = \begin{cases} 0 & \text{if } i > n \\ LAS^+(i+1,j) & \text{if } i \le n \text{ and } A[i] \le A[j] \\ \max\left\{LAS^+(i+1,j), 1 + LAS^-(i+1,i)\right\} & \text{otherwise} \end{cases}$$

$$LAS^-(i,j) = \begin{cases} 0 & \text{if } i > n \\ LAS^-(i+1,j) & \text{if } i \le n \text{ and } A[i] \ge A[j] \\ \max\left\{LAS^-(i+1,j), 1 + LAS^+(i+1,i)\right\} & \text{otherwise} \end{cases}$$

Solution:

- two for loops:
  - $i \leftarrow n$ down to 1
  - $j \leftarrow i-1$ down to 1
- for $i = 1 \to n$ return $max(LAS^+(i+1,1), LAS^-(i+1,1))$. This problem is slightly harder than the others in the time required so we'll accept anything of the form: $max(LAS^+(n,n), LAS^-(n,n))$

■

b. **BYC & BYE**

Given an array $A[1..n]$ of integers, compute the length of a **longest decreasing subsequence**. Let $LDS(i,j)$ denote the length of the longest decreasing subsequence of $A[i..n]$ where every element is smaller than $A[j]$.

$$LDS(i,j) = \begin{cases} 0 & \text{if } i > n \\ LDS(i+1,j) & \text{if } i \leq n \text{ and } A[j] \leq A[i] \\ \max\{LDS(i+1,j), 1+LDS(i+1,i)\} & \text{otherwise} \end{cases}$$

Solution:

- two for loops:
  - $i \leftarrow n$ down to 1
  - $j \leftarrow i-1$ down to 0

  We won't make off for ± 1 issues.
- return $A[1,0]$.

■

c. **BYD & BYG**

Given an array $A[1..n]$, compute the length of a longest **palindrome** subsequence of A. Let $LPS(i,j)$ denote the length of the longest palindrome subsequence of $A[i..j]$.

$$LPS(i,j) = \begin{cases} 0 & \text{if } i > j \\ 1 & \text{if } i = j \\ \max\begin{Bmatrix} LPS(i+1,j) \\ LPS(i,j-1) \end{Bmatrix} & \text{if } i < j \text{ and } A[i] \neq A[j] \\ \max\begin{Bmatrix} 2+LPS(i+1,j-1) \\ LPS(i+1,j) \\ LPS(i,j-1) \end{Bmatrix} & \text{otherwise} \end{cases}$$

Solution:

- two for loops:
  - $i \leftarrow n$ down to 1
  - $j \leftarrow i+1$ down to n

  We won't make off for ± 1 issues.
- return $A[1,n]$.

■

d. **BYB & BYF**

Given an array $A[1..n]$ of integers, compute the length of a longest **convex** subsequence of A. Let $LCS(i, j)$ denote the length of the longest convex subsequence of $A[i..n]$ whose first two elements are $A[i]$ and $A[j]$.

$$LCS(i,j) = 1 + \max\{LCS(j,k) | j < k \leq n \text{ and } A[i] + A[k] > 2A[j]\}$$

Solution:

- three for loops:
  - $i \leftarrow n-1$ down to 1
  - $j \leftarrow n$ down to $i+1$
  - $j \leftarrow j+1$ to n

  We won't make off for ± 1 issues. Mainly want to see three for loops.
- return $max(A[1..n-1, i+1..n])$.Will give full credit to anyone that take the max of the first two dimensions.

■