Note: There is a lot of confusion on how to write English descriptions of recurrences. I wrote two such descriptions in the first two solutions of Problem 1. The rest I'm keeping brief so you can practice describing solutions.

Describe recursive backtracking algorithms for the following problems. *Don't worry about running times.*

1. Given an array $A[1..n]$ of integers, compute the length of a ***longest increasing subsequence***.

Solution: Add a sentinel value $A[0] = -\infty$. Let $LIS(i, j)$ denote the length of the longest increasing subsequence of $A[i..n]$ where every element is larger than $A[j]$. This function obeys the following recurrence:

$$LIS(i,j) = \begin{cases} 0 & \text{if } i > n & \text{(1a)} \\ LIS(i+1, j) & \text{if } i \le n \text{ and } A[j] \ge A[i] & \text{(1b)} \\ \max \left\{ \begin{array}{c} LIS(i+1, j) \\ 1 + LIS(i+1, i) \end{array} \right\} & \text{otherwise} & \text{(1c)} \end{cases}$$

$LIS(i, j)$ denotes the length of the longest increasing subsequence of $A[i..n]$ where every element is larger than $A[j]$. The cases are described as follows:

- 1a is the base case. If $i > n$ there is nothing in the array's suffix and therefore, nothing to put in the subsequence.
- 1b if $A[j] \ge A[i]$, then you simply cannot add $A[i]$ to the subsequence and thus, the LIS of $A[i..n]$ assuming all values must be larger than $A[j]$ is simply the LIS of $A[i+1..n]$ assuming all valuues must be larger than $A[j]$.
- 1c if $A[j] < A[i]$, then $A[i]$ can be added to the LIS, but that doesn't mean it should be. Hence we need to compare the case where we don't add $A[i]$ to the LIS ($LIS(i+1, j)$) to the case where we do add $A[i]$ to the LIS ($1 + LIS(i+1, i)$).

We need to compute $LIS(1, 0)$ to get the length of the LIS for the total sequence. ■

Solution: Let $LIS(i)$ denote the length of the longest increasing subsequence of $A[i\,..\,n]$ that begins with $A[i]$. This function obeys the following recurrence:

$$LIS(i) = \begin{cases} 1 & \text{if } A[j] \le A[i] \text{ for all } j > i \quad \text{(2a)} \\ 1 + \max\left\{LIS(j) \mid j > i \text{ and } A[j] > A[i]\right\} & \text{otherwise} \quad \text{(2b)} \end{cases}$$

The cases can be described as follows:

- 2a is the base case. If $A[j] \le A[i]$, it means that $A[j]$ cannot be a part of the longest LIS that begins with $A[i]$. Hence, the maximum LIS that begins with $A[i]$ assuming $A[j]$ is in the LIS must be 1.
- 2b simply looks at all the possible elements after $A[i]$ and says, "if this element must be a part of the subsequence, what is the LIS that starts with $A[i]$?"

(The first case is actually redundant if we define $\max\varnothing = 0$.) We need to compute $\max_i LIS(i)$. ■

Solution: Add a sentinel value $A[0] = -\infty$. Let $LIS(i)$ denote the length of the longest increasing subsequence of $A[i\,..\,n]$ that begins with $A[i]$. This function obeys the following recurrence:

$$LIS(i) = \begin{cases} 1 & \text{if } A[j] \le A[i] \text{ for all } j > i \\ 1 + \max\left\{LIS(j) \mid j > i \text{ and } A[j] > A[i]\right\} & \text{otherwise} \end{cases}$$

(The first case is actually redundant if we define $\max\varnothing = 0$.) We need to compute $LIS(0) - 1$; the -1 removes the sentinel $-\infty$ from the start of the subsequence. ■

Solution: Add sentinel values $A[0] = -\infty$ and $A[n+1] = \infty$. Let $LIS(j)$ denote the length of the longest increasing subsequence of $A[1\,..\,j]$ that ends with $A[j]$. This function obeys the following recurrence:

$$LIS(j) = \begin{cases} 1 & \text{if } j = 0 \\ 1 + \max\left\{LIS(i) \mid i < j \text{ and } A[i] < A[j]\right\} & \text{otherwise} \end{cases}$$

We need to compute $LIS(n+1) - 2$; the -2 removes the sentinels $-\infty$ and ∞ from the subsequence. ■

2. Given an array $A[1..n]$ of integers, compute the length of a ***longest decreasing subsequence***.

Solution: Add a sentinel value $A[0] = \infty$. Let $LDS(i, j)$ denote the length of the longest decreasing subsequence of $A[i..n]$ where every element is smaller than $A[j]$. This function obeys the following recurrence:

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

We need to compute $LDS(1, 0)$. ■

Solution: Multiply every element of A by -1, and then compute the length of the longest increasing subsequence using the algorithm from problem 1. ■

3. Given an array $A[1..n]$ of integers, compute the length of a ***longest alternating subsequence***.

Solution: We define two functions:

- 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.
- Let $LAS^-(i,j)$ denote the length of the longest alternating subsequence of $A[i..n]$ whose first element (if any) is smaller than $A[j]$ and whose second element (if any) is larger than its first.

These two functions satisfy the following mutual recurrences:

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

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

To simplify computation, we consider two different sentinel values $A[0]$. First we set $A[0] = -\infty$ and let $\ell^+ = LAS^+(1,0)$. Then we set $A[0] = +\infty$ and let $\ell^- = LAS^-(1,0)$. Finally, the length of the longest alternating subsequence of A is $\max\{\ell^+, \ell^-\}$. ■

Solution: We define two functions:

- Let $LAS^+(i)$ denote the length of the longest alternating subsequence of $A[i..n]$ that starts with $A[i]$ and whose second element (if any) is larger than $A[i]$.
- Let $LAS^-(i)$ denote the length of the longest alternating subsequence of $A[i..n]$ that starts with $A[i]$ and whose second element (if any) is smaller than $A[i]$.

These two functions satisfy the following mutual recurrences:

$$LAS^+(i) = \begin{cases} 1 & \text{if } A[j] \le A[i] \text{ for all } j > i \\ 1 + \max\left\{LAS^-(j) \;\middle|\; j > i \text{ and } A[j] > A[i]\right\} & \text{otherwise} \end{cases}$$

$$LAS^-(i) = \begin{cases} 1 & \text{if } A[j] \ge A[i] \text{ for all } j > i \\ 1 + \max\left\{LAS^+(j) \;\middle|\; j > i \text{ and } A[j] < A[i]\right\} & \text{otherwise} \end{cases}$$

We need to compute $\max_i \max\{LAS^+(i), LAS^-(i)\}$. ■

To think about later:

4. Given an array $A[1..n]$ of integers, compute the length of a longest ***convex*** subsequence of A.

Solution: Let $LCS(i, j)$ denote the length of the longest convex subsequence of $A[i..n]$ whose first two elements are $A[i]$ and $A[j]$. This function obeys the following recurrence:

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

Here we define $\max \varnothing = 0$; this gives us a working base case. The length of the longest convex subsequence is $\max_{1\leq i<j\leq n} LCS(i, j)$. ■

Solution (with sentinels): Assume without loss of generality that $A[i] \geq 0$ for all i. (Otherwise, we can add $|m|$ to each $A[i]$, where m is the smallest element of $A[1..n]$.) Add two sentinel values $A[0] = 2M + 1$ and $A[-1] = 4M + 3$, where M is the largest element of $A[1..n]$.

Let $LCS(i, j)$ denote the length of the longest convex subsequence of $A[i..n]$ whose first two elements are $A[i]$ and $A[j]$. This function obeys the following recurrence:

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

Here we define $\max \varnothing = 0$; this gives us a working base case.

Finally, we claim that the length of the longest convex subsequence of $A[1..n]$ is $LCS(-1, 0) - 2$.

Proof: First, consider any convex subsequence S of $A[1..n]$, and suppose its first element is $A[i]$. Then we have $A[-1] - 2A[0] + A[i] = 4M + 3 - 2(2M + 1) + A[i] = A[i] + 1 > 0$, which implies that $A[-1] \cdot A[0] \cdot S$ is a convex subsequence of $A[-1..n]$. So the longest convex subsequence of $A[1..n]$ has length at most $LCS(-1, 0) - 2$.

On the other hand, removing $A[-1]$ and $A[0]$ from any convex subsequence of $A[-1..n]$ laves a convex subsequence of $A[1..n]$. So the longest subsequence of $A[1..n]$ has length at least $LCS(-1, 0) - 2$. □

■

5. Given an array $A[1..n]$ of integers, compute the length of a longest ***weakly-increasing*** subsequence of A.

Solution: We use the approach used in the solution for the longest convex subsequence problem in the lab on 3/4 to solve this problem. Let $LWIS(i, j)$ denote the length *minus* 1 of a longest weakly increasing subsequence of a *nonempty* array $A[i..n]$ whose first element is $A[i]$ and whose second element (if any) is $A[j]$. $LWIS$ obeys the following recurrence:

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

Here we define $\max \varnothing = 0$; this gives us a working base case. The final answer is given by

$$\mathrm{LWIS}(A[1..n]) = \begin{cases} 0 & \text{if } n = 0 \\ 1 + \max\limits_{1 \leq i < j \leq n} LWIS(i,j) & \text{otherwise} \end{cases}.$$

■

6. Given an array $A[1..n]$, compute the length of a longest ***palindrome*** subsequence of A.

Solution: Let $LPS(i,j)$ denote the length of the longest palindrome subsequence of $A[i..j]$. This function obeys the following recurrence:

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

We need to compute $LPS(1,n)$. ■

Solution: Let $LPS(i,j)$ denote the length of the longest palindrome subsequence of $A[i..j]$. Before stating a recurrence for this function, we make the following useful observation.[a]

Claim 1. *If $i < j$ and $A[i] = A[j]$, then* $\mathrm{LPS}(i,j) = 2 + \mathrm{LPS}(i+1,j-1)$.

Proof: Suppose $i < j$ and $A[i] = A[j]$. Fix an arbitrary longest palindrome subsequence S of $A[i..j]$. There are four cases to consider.

- If S uses neither $A[i]$ nor $A[j]$, then $A[i] \bullet S \bullet A[j]$ is a palindrome subsequence of $A[i..j]$ that is longer than S, which is impossible.
- Suppose S uses $A[i]$ but not $A[j]$. Let $A[k]$ be the last element of S. If $k = i$, then $A[i] \bullet A[j]$ is a palindrome subsequence of $A[i..j]$ that is longer than S, which is impossible. Otherwise, replacing $A[k]$ with $A[j]$ gives us a palindrome subsequence of $A[i..j]$ with the same length as S that uses both $A[i]$ and $A[j]$.
- Suppose S uses $A[j]$ but not $A[i]$. Let $A[h]$ be the first element of S. If $h = j$, then $A[i] \bullet A[j]$ is a palindrome subsequence of $A[i..j]$ that is longer than S, which is impossible. Otherwise, replacing $A[h]$ with $A[i]$ gives us a palindrome subsequence of $A[i..j]$ with the same length as S that uses both $A[i]$ and $A[j]$.
- Finally, S might include both $A[i]$ and $A[j]$.

In all cases, we find either a contradiction or a longest palindrome subsequence of $A[i..j]$ that uses both $A[i]$ and $A[j]$. □

Claim 1 implies that the function LPS satisfies the following recurrence:

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

We need to compute $LPS(1,n)$. ■

[a] And yes, optimizations like this require a proof of correctness, both in homework and on exams. Premature optimization is the root of all evil.