L19

Problem 1 (Parse the Algorithm) *You are interested in finding the* ***Longest Increasing Subsequence*** *in an array of values. In the context of the book sorting problem, you want to find the largest set of books that get taller as they go from left to right, though the books do not necessarily need to be adjacent. Here is the pseudocode provided:*

```
1: procedure LIS(A)
2:     n ←len(A)
3:     opt ← array of length n
4:     opt[0] = 1
5:     for j in range(1, n) do
6:         opt[j] ← 1 + max(opt[i]: i < j, A[i] < A[j])        ▷ The max evaluates to 0 if no terms qualify.
7:     end for
8:     return max(opt)
9: end procedure
```

In this problem, you will be parsing this pseudocode.

(a) For the following set of book heights, fill in the table below.

j	0	1	2	3	4	5	6	7	8	9	10
A[j]	5	2	8	6	3	6	9	7	5	4	3
OPT[j]											

(b) In the previous problem, we defined OPT(t) to mean the value of the best schedule possible that ends by time t. What does OPT(j) represent here?

(c) In the previous problem, at each value of t we were deciding whether to take the class or not. In this problem, what decision are we making at each value of i?

(d) In the previous problem, we set opt[0]=0. Why do we set it to be 1 here?

(e) In the previous problem, we returned opt[n]. Why don't we do that here?

Problem 2 (Edit Distance) *You want to change your shop sign from VANILLA to SALSA.*

(a) *Find the edit distance between these two words. What is the alignment of the two words that minimizes the edit distance?*

(b) *Following the pseudocode given in the slide, fill out the edit distance table below to compare VANILLA to SALSA.*

		S	A	L	S	A
V						
A						
N						
I						
L						
L						
A						

(c) *How can we find the alignment of words from this table?*