COMP 251 Assignment 3 - Written

3)

Given an array of integers A, there can be 3 cases regarding implementing active lists with A[i]:

1. A[i] is the smallest, so we create a new list of length 1
2. A[i] is the largest, so we clone existing list and add A[i] to end
3. A[i] is in the middle, so we find the list with the largest end element thats smaller than A[i], clone and add A[i] to end and we discard lists with the same length as our new list

Note: Here, we have the end element of smaller lists always smaller than the end element of larger lists

```
//Pseudocode
longestIncrementalSubseunce(int[] s)

    // Keep track of predecessors and ends of each subsequence
    int[] preds = new int[s.length]
    int[] ends = new int[s.length + 1]

    // Start off with 0 length
    int length = 0;

    for i = 0 to s.length - 1
        // Binary search
        int low = 1
        int high = length

        while low <= high
            int middle = Ceil((low + high) / 2)

            if s[ends[middle] < s[i]]
                low = middle + 1
            else
                high = middle - 1

        // Update preds, and replace ends
```

```
            preds[i] = ends[low]
            ends[low] = i

            // Update length
            if (low > length)
                  length = low

      // Get longest increment subsequence
      int[] longestIncSub = new int[length]
      int j = ends[length]

      for i = length - 1 to 0
            longestIncSub[i] = s[j]
            j = preds[j]

      return longestIncSub
```

Binary search time is O(log n), which runs in a loop of n times.
Therefore, run time is O(n log n)

(4)

match = 1
mismatch = -1
insertion/deletion = -1

Build a table

	-	A	A	C	T
-	0	-1	-2	-3	-4
C	-1	-1	-2	-3	-4
A	-2	0	0	-1	-2
T	-3	-1	-1	-1	0

- go through the values and pick best result from 3 cases which are labelled with arrows

↓ Trace back

	-	A	A	C	T
-	0	-1	-2	-3	-4
C	-1	-1	-2	-3	-4
A	-2	0	0	-1	-2
T	-3	-1	-1	-1	0

{ A A C T
{ C A - T

horizontal arrow, so there is a gap

∴ This is the optimal sequence alignment, so the max score = [0]