Jordan Allard
CSCI 264-01
Homework 4

HW4 Problem 5 Writeup

a) Heart of the Solution

I) WHAT: S[i, j] = the length of the longest convex subsequence chosen from the first j elements and ending with the ith and jth element

II) HOW:

```
S[i, j] = -infinity     if i > j
S[i, j] = max( S[k, i] ) + 1, where k = 1 to i - 1,
      if a[k] + a[j] >= 2*a[i], otherwise:
S[i, j] = 0
```

III) WHERE: max(S[k, n]) where k = 1 to n - 1

b) Pseudocode

```
Given an array A of ints of length n
For i = 1 to n:
      For j = 1 to n:
            If i > j: S[i, j] = -infinity
maxLength = 0
For j = 1 to n:
      For i = 1 to j:
            S[i, j] = 0
            For k = 1 to i - 1:
                  If a[k] + a[j] >= 2*a[i]:
                        If S[k, i] == 0 and S[i, j] < 3:
                              S[i, j] = 3
                        Else if S[k, i] + 1 > S[i, j]:
                              S[i, j] = S[k, i] + 1
            If S[i, j] > maxLength: maxLength = S[i, j]
Return maxLength
```

c) Proof of Correctness (Explanation of HOW)

To find the maximum sequence that ends with i and j, the algorithm should find the maximum sequence that ends with i and add j to it if it satisfies the convex condition. By checking the length of every possible sequence that ends with i and taking the maximum sequence that validates the convex condition, the algorithm is guaranteed to find the longest sequence for each combination of i and j. Since the algorithm iterates through every combination of i and j, it will always find the longest possible convex sequence.

d) Running Time Estimate

O(n^3)

e) Running Time Estimate Reasoning

The main loop of this algorithm is a double nested for loop, where each level of the loop runs from 1 to the previous index. Since the highest level of the loop runs n times, the worst case scenario will have a running time of n x n x n = O(n^3). Thus, the total running time, including the time taken for input and initializing values, is O(n) + O(n^2) + O(n^3) = O(n^3).