CS3343 Analysis of Algorithms
Homework 1

Justify all of your answers with comments/text in order to receive full credit. Completing the assignment in LaTeXwill earn you extra credit on the Midterm.

1. Longest Sorted Subarray (8 points)

1	2	3	4	5	6	7	8	9	10
2	4	3	8	5	6	7	9	0	1

3

2

Consider the following problem:

Input: An array $A[1\dots n]$ of integers
Output: The largest integer m such that the array $A[1\dots n]$ has subarray of length m which is in sorted order (i.e, increasing order).

//As we can see 4 is largest subarray in ascending order

The following pseudocode finds the length of the longest of the given array $A[1\dots n]$ by considering all possible subarrays:

Algorithm 1 longestSubArray(int $A[1\dots n]$)

```
k = n;  ——— 1 time
while ( true ) do
    //(I) The longest increasing subarray of A has length ≤ k
    low = 1;
    high = k;
    while ( high ≤ n ) do
        if ( isIncreasing(A[low ... high]) ) then
            return k;
        end if
        low++;
        high++;
    end while
    k--;
end while
```

n= 10

k	high	low
10	10	1
9	11	2
	9	1

//Algorithm removes elements from rear until we find largest subArray in ascending order

The following code checks if an array is increasing (i.e., each number is smaller than the next in the array).

// cycle through Array looking for largest subArray in
//ascending order

Algorithm 2 isIncreasing(int $C[a\dots b]$)

```
i = a;  —— 1 time
while i < b do
    if ( C[i] ≥ C[i+1] ) then
        return false; // Found pair out-of-order
    end if
    i++;
end while
return true; // No pairs were out-of-order  —— 1 time
```

4	3

return false

//Think we actually stay here for a while ?

Example: longestSubArray($[2, 4, 3, 8, 5, 6, 7, 9, 0, 1]$) returns 4
Justification: [~~2, 4, 3, 8~~, 5, 6, 7, 9, ~~0, 1~~] = [5, 6, 7, 9] which is a longest increasing subarray of the original array.

(1) (2 points) Consider running longestSubArray on the array: => returns 5

$[119, \underbrace{100, \overbrace{112, 114, 125}^{3}}_{4}, 113, \underbrace{110, \overbrace{129, 130, 140, 142}^{4}}_{5}, 115, 120]$ Sub Array: [110, 129, 130, 140, 142]

What does longestSubArray return ~~and~~ what is the longest sorted subarray of A?

(2) (4 points) Use induction to prove the loop invariant (I) is true and then use this to prove the correctness of the algorithm. Specifically complete the following:

(a) Base case

(b) Inductive step

(c) Termination step

(**Hint:** the outer loop never terminates but consider what can you say about the k value that causes us to return.)

(3) (1 point) Give the best-case runtime of longestSubArray in asymptotic (i.e., O) notation as well as a description of an array which would cause this behavior.

(4) (1 point) Give the worst-case runtime of longestSubArray in asymptotic (i.e., O) notation as well as a description of an array which would cause this behavior.

(5) (0 points) Is this an efficient algorithm for finding the longest sorted subarray? Can you find a better algorithm for computing this?

2. Asymptotic Notation (4 points)

Show the following using the definitions of O, Ω, and Θ.

(1) (2 points) $2n^3 + n^2 + 4 \in \Theta(n^3)$

(2) (2 points) $3n^4 - 9n^2 + 4n \in \Theta(n^4)$

(**Hint:** careful with the negative number)

3. Summations (4 points)

Find the order of growth of the following sums.

(1) $\sum\limits_{i=10}^{n}(5i+3)$

(2) $\sum\limits_{i=0}^{\log_2(n)} 2^i$ (for simplicity you can assume n is a power of 2)

4. Master theorem (4 points)

Use the master theorem to find tight asymptotic bounds for the following recurrences. Justify your answers.

(1) $T(n) = 6T(n/6) + n$

(2) $T(n) = 9T(n/3) + \sqrt{n}$

(3) $T(n) = T(n/2) + T(n/2) + n$

(4) $T(n) = T(2n/3) + n$