CS3343 Analysis of Algorithms Fall 2019

Homework 1

Due 9/5/19 before 11:59pm

Justify all of your answers with comments/text in order to receive full credit. Completing the assignment in LaTeXwill earn you extra credit on Midterm 1.

1. Longest Increasing Subsequence (8 points)

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 subsequence of length m which is strictly increasing.

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

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

```
k = 0;
while (true) do
    //(I) The longest increasing subsequence of A has length ≤ n − k
    do
        Let B = A with k of its elements removed
        if ( isIncreasing(B) ) then
            return n − k;
        end if
    while ( there are other untested choices of k elements to remove )
    k++;
end while
```

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

Algorithm 2 isIncreasing(int $C[1 \dots n]$)

```
i = 1;
while i < n do
    if ( C[i] ≥ C[i + 1] ) then
        return false;
    end if
    i++;
end while
return true;
```

Example: longestSubSeq($[2, 4, 3, 8, 5, 5, 6, 7, 9]$) returns 6
Justification: $[2, 4, \cancel{3}, \cancel{8}, 5, \cancel{5}, 6, 7, 9] = [2, 4, 5, 6, 7, 9]$ which is a longest increasing subsequence of the original array.

(1) (2 points) Consider running longestSubSeq on the array:

$$[1\cancel{1}9, 100, 112, 1\cancel{25}, 113, 1\cancel{1}0, 1\cancel{30}, 115, 120] = \boxed{100 \quad 112 \quad 113 \quad 115 \quad 120}$$

What does longestSubSeq return and what array B causes this return?

(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
  (**Hint:** you can assume that line 5 will successively try all possible choices of how to remove k elements from A.)
  (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 longestSubSeq 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 longestSubSeq in asymptotic (i.e., O) notation as well as a description of an array which would cause this behavior.

2. Asymptotic Notation (8 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) (4 points) Suppose $f(n) \in O(g_1(n))$ and $f(n) \in O(g_2(n))$. Which of the following are true? Justify your answers using the definition of O. Give a counter example if it is false.
  (a) $f(n) \in O(\ 5 \cdot g_1(n) + 100\ )$
  (b) $f(n) \in O(\ g_1(n) + g_2(n)\ )$
  (c) $f(n) \in O(\ \frac{g_1(n)}{g_2(n)}\ )$
  (d) $f(n) \in O(\ \max(g_1(n), g_2(n))\ )$

3. Summations (4 points)

Find the order of growth of the following sums.

(1) $\sum_{i=10}^{n}(5i+3)$. $\sum_{i=10}^{n} 5i + \sum_{i=10}^{n} 3 = \sum_{i=10}^{n} 5i + 3n - 27 = 5\sum_{i=10}^{n} i + 3n - 27 = 5\left(\frac{n(n+1)}{2}\right) + 3n - 27$
$= \frac{5n^2 + 5n}{2} + 3n - 27$ $\boxed{\text{So, } \sum_{i=10}^{n} 5i+3 \in \Theta(n^2)}$

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

$2^0 + 2^1 + 2^2 + \ldots \ldots 2^{\log n}$ Capes

$\boxed{\text{So, } \sum_{i=0}^{\log_2 n} 2^i \in \Theta(n)}$

① Longest Increasing Subsequence.

Base case: $K=0$

(I) The longest increasing sequence of A has length $\leq n-K$

is true because each element is increasing sequence.

inductive step: assume A has length $\leq n-K$ we need to show $LSA(A) \neq n-K$

Prove $A[i] \leq n-K+1$ — decreasing

If solution was $n-K$ then one of the B array that we tried would have been increasing since line 5 tries all B arrays of length $n-K$

Termination step: the program terminates. when all element are tested. and finds the longest sequence until the last element.

best case: best case would be $O(n^2)$ in the given array is already sorted in increasing order.

worst case: $O(2^n)$ if the array is sorted in decending order.

(2) Asymptotic Notation

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

$f(n) \le c \cdot g(n) \ \forall \ n \le n_0$

$2n^3+n^2+4 \le 2n^3+n^3+4n^3$

$2n^3+n^3+4 \le 7n^3$

$c=7 \quad n_0=1$

$f(n)\ O(n^3)$

$f(n) \ge c \cdot \Omega\, g(n) \ \forall \ n \ge n_0$

$2n^3+n^3+4n^3 \ge 2n^3+0+0$

$\ge 2n^3$

$c=2 \quad n_0=1$

$f(n)\ \Omega(n^3)$

So, $f(n)\ \Theta(n^3)$

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

$f(n) \le g(n) \ \forall \ n \le n_0$

$3n^4-9n^2+4n \le 3n^4$

$c=3 \quad n_0=1$

$f(n) \ge c \cdot \Omega\, g(n) \ \forall \ n \ge n_0$

$3n^4-9n^2+4n \ge 3n$

$c=3$

$n_0=1$

$f(n)\ \Omega(n^4)$

So, $f(n)\ \Theta(n^4)$

(3a) $f(n) \in O(5g_1(n)+100)$

True

we know $f(n) \in O(g_1(n))$

So $0 \le f(n) \le c \cdot g_1(n)$ for $n \ge n_0$

(3b) $f(n) \in O(g_1(n) + g_2(n))$

True:

$f(n) \in O(g_1(n))$

$f(n) \in O(g_2(n))$

$f(n) \leq c_1 \cdot (g_1(n) + g_2(n))$

$f(n) \leq c_2 \cdot g_1(n) + g_2(n))$ for all $n \geq n_1$

(3c) $f(n) \in O\left(\frac{g_1(n)}{g_2(n)}\right)$

False:

$n \in O(n)$ $n^2 \in O(n^2)$ but $(n^2 \cdot n^2) \notin O(n^2)$

(3d) $f(n) \in O(\max g_1(n)\ g_2(n))$

True

$f(n) \in O(g_1(n)) \Rightarrow f(n) \leq c \cdot (g_1(n)) \to n \geq n_1$

$f(n) \in O(g_2(n)) \Rightarrow f(n) \leq c \cdot (g_2(n))$

$f(n) \in O(\max(g_1(n), g_2(n)) \Rightarrow c \cdot \max(g_1(n)\ g_2(n))$

for all $n \geq n_0$

and $c \geq 1$