CSC 212: Data Structures and Abstractions

Spring 2019

University of Rhode Island

Weekly Problem Set #3

Due Thursday 2/21 at the beginning of class. Please turn in neat, and organized, answers **hand-written** on standard-sized paper **without any fringe**. At the top of each sheet you hand in, please write your name, and ID.

1. Prove the following. Additionallly, is there a more accurate Big Omega or Big O that could be used?

   - $10n^2 = \Omega(n)$
     $(10n^2 >= n >= 0)\forall(n > 1)$
     Accurate:$\Omega(n^2)$
   - $1 = O(n)$
     $(1 <= n)\forall(n > 1)$
     Accurate:$O(1)$
   - $4n^3 + 3n + 2 = O(n^3)$
     $(4n^3 + 3n + 2 <= 5n^3)\forall(n >= 2)$
   - $n \log n = \Omega(n \log n)$
     $(n \log n >= n \log n)\forall(n >= 1)(0 < c <= 1)$
   - $n^4 = O(2^n)$
     $(n^4 <= 2^n)\forall(n >= 16)$
     Accurate:$O(n^4)$

2. Mark each of the following as true or false.

T(n)	Big O	T/F	Big Omega	T/F	Big Theta	T/F
$n^2/10 + 10n \log n$	$O(n \log n)$	F	$\Omega(n \log n)$	T	$\Theta(n \log n)$	F
$2n^2 + n \log n$	$O(n^2)$	T	$\Omega(n)$	T	$\Theta(\log n)$	F
$(n/2)logn + 4n$	$O(2^n)$	T	$\Omega(n \log n)$	T	$\Theta(n \log n)$	T
$10\sqrt{n} + 2 \log n$	$O(\log n)$	F	$\Omega(n)$	F	$\Theta(\log n)$	F
$3\sqrt{n} + 10 \log n$	$O(\sqrt{n})$	T	$\Omega(1)$	T	$\Theta(\sqrt{n})$	T

3. Complete the following table.

T(n)	Big Theta
$\log n + 200n \log n$	$\Theta(n \log n)$
$2^n + n^2$	$\Theta(2^n)$
$\sqrt{n} + \log n$	$\Theta(\sqrt{n})$
$2n + 3n + 4n + 5n + 6n$	$\Theta(n)$
$\sqrt{n} + 10 \log n$	$\Theta(\sqrt{n})$
$200n * 10n + \log n$	$\Theta(n^2)$

4. Complete the following table using Big Θ notation, measuring performance by the number of comparisons.

Algorithm	Best Case	Average Case	Worst Case
Selection Sort	$\Theta(n^2)$	$\Theta(n^2)$	$\Theta(n^2)$
Insertion Sort	$\Theta(n)$	$\Theta(n^2)$	$\Theta(n^2)$
Bubble Sort	$\Theta(n)$	$\Theta(n^2)$	$\Theta(n^2)$
Maximum of an Unsorted Array	$\Theta(n)$	$\Theta(n)$	$\Theta(n)$
Median of a Sorted Array	$\Theta(1)$	$\Theta(1)$	$\Theta(1)$
Mode of a Sorted Array	$\Theta(n)$	$\Theta(n)$	$\Theta(n)$

5. Given the array `A` with elements $[22, 15, 36, 44, 10, 3, 9, 13, 29, 25]$, illustrate the performance of the selection-sort algorithm from the lecture slides on `A`. To illustrate the performance, depict the status of the array after line 15 at every iteration.

   (i:0) $[3, 15, 36, 44, 10, 22, 9, 13, 29, 25]$

   (i:1) $[3, 9, 36, 44, 10, 22, 15, 13, 29, 25]$

   (i:2) $[3, 9, 10, 44, 36, 22, 15, 13, 29, 25]$

   (i:3) $[3, 9, 10, 13, 36, 22, 15, 44, 29, 25]$

   (i:4) $[3, 9, 10, 13, 15, 22, 36, 44, 29, 25]$

   (i:5) $[3, 9, 10, 13, 15, 22, 36, 44, 29, 25]$

   (i:6) $[3, 9, 10, 13, 15, 22, 25, 44, 29, 36]$

   (i:7) $[3, 9, 10, 13, 15, 22, 25, 29, 44, 36]$

   (i:8) $[3, 9, 10, 13, 15, 22, 25, 29, 36, 44]$

6. Given the array `A` with elements $[22, 15, 36, 44, 10, 3, 9, 13, 29, 25]$, illustrate the performance of the insertion-sort algorithm on `A`. Again, use the function provided in the lecture notes, and depict the status of the array after line 14 at every iteration. (Line 14 signifies the moment after the if statement terminates)

(i:0) $[22, 15, 36, 44, 10, 3, 9, 13, 29, 25]$

(i:1) $[15, 22, 36, 44, 10, 3, 9, 13, 29, 25]$

(i:2) $[15, 22, 36, 44, 10, 3, 9, 13, 29, 25]$

(i:3) $[15, 22, 36, 44, 10, 3, 9, 13, 29, 25]$

(i:4) $[10, 15, 22, 36, 44, 3, 9, 13, 29, 25]$

(i:5) $[3, 10, 15, 22, 36, 44, 9, 13, 29, 25]$

(i:6) $[3, 9, 10, 15, 22, 36, 44, 13, 29, 25]$

(i:7) $[3, 9, 10, 13, 15, 22, 36, 44, 29, 25]$

(i:8) $[3, 9, 10, 13, 15, 22, 29, 36, 44, 25]$

(i:9) $[3, 9, 10, 13, 15, 22, 25, 29, 36, 44]$

7. What type of array would yield the worst-case performance for an insertion sort array? What about the best-case performance?

   The worst case input would be an array in reverse order as it would give the largest distance for each element in the array to move.
   The best case would be an array already sorted; insertion sort wouldn't have to do any swaps

8. Can you improve the worst-case performance of selection sort? Why or why not?

   You cannot improve the performance of selection sort. It has to always traverse the entire array for each element in the list in order to find the nth smallest/largest term.

9. How many inversions are present in each of the following arrays?

   A: [1, 5, 4, 3, 3, 7] $->$ 5

   B: [5, 4, 3, 2, 1] $->$ 10

   C: [1, 2, 3, 4, 5] $->$ 0

   D: [5, 1, 3, 2, 4] $->$ 5

   E: [6, 9, 1, 4, 10] $->$ 4