Assignment - 1

Total Points: 100

Instructions

Submission

You can either write your answer using Word/LaTex or take a picture/scan of your hand-written(neat and tidy) solution, then put your solution in one PDF file. Submit your PDF online using Blackboard. All submission must be posted before the deadline.

Problem 1 (Max Points:10)

Use mathematical induction to show that the solution to the recurrence:

$$T(n) = \begin{cases} 2 & \text{if } n = 2 \\ 2T(\frac{n}{2}) + n & n = 2^k, k > 1. \end{cases}$$

is $T(n) = n \lg n$.

Problem 2 (Max Points: 30)

Indicate for each pair of expressions (A, B) whether A is big O, little o, big Ω, little *omega*, or Θ of B. The expressions involve some constants; assume $k >= 1$, $\epsilon > 0$, and $c > 1$ are constants. Justify your answer with proper reasoning.

	A	B	O	o	Ω	ω	Θ
a.	$\lg^k n$	n^ϵ					
b.	n^k	c^n					
c.	$\sqrt{n}$	$n \sin n$					
d.	2^n	$2^{n/2}$					
e.	$n^{\lg c}$	$c^{\lg n}$					
f.	$\lg(n!)$	$\lg(n^n)$					

Problem 3 (Max Points:10)

We can express insertion sort as a recursive procedure as follows. In order to sort A[1..n], we recursively sort A[1...n-1] and then insert A[n] into the sorted array A[1...n-1].Write a recurrence for the running time of this recursive version of insertion sort and also solve it.

Problem 4 (Max Points:10)

Use the following ideas to develop a non-recursive, linear time algorithm for the max-subarray problem. Start at the left end of the array, and progress toward the right, keeping track of the maximum subarray seen so far. Knowing a maximum subarray of $A[1..j]$, extend the answer to find a maximum subarray for index $j+1$ by using the following observation: a maximum subarray of $A[1..j+1]$ is either a maximum subarray of $A[1..j]$ or a subarray $A[i..j+1]$, for some $1 \leq i \leq j+1$. Determine a maximum subarray of the form $A[i..j+1]$ in constant time based on knowing a maximum subarray ending at index j.

Problem 5 (Max Points:10)

Use a recursion tree to determine a good asymptotic upper bound on the recurrence $T(n) = 4T(n/2) + n$. Use the substitution method to verify your answer.

Problem 6 (Max Points: 30)

For each of the following recurrences, give an expression for the run time T(n) if the recurrence can be solved with the Master Theorem. Otherwise, indicate that the Master Theorem does not apply. Justify your answer with reasoning.

(a) $T(n) = 2T(n/2) + n^4$

(b) $T(n) = T(7n/10) + n$

(c) $T(n) = 16T(n/4) + n^2$

(d) $T(n) = 2T(n/4) + \sqrt{n}$

(e) $T(n) = \sqrt{2}T(n/2) + \log n$

(f) $T(n) = 64T(n/8) - n^2 \log(n)$

(g) $T(n) = 2T(n/4) + n^{0.51}$

(h) $T(n) = 16T(n/4) + n!$

(i) $T(n) = 0.5T(n/2) + 1/n$

(j) $T(n) = 2^n T(n/2) + n^n$