KING SAUD UNIVERSITY

COLLEGE OF COMPUTER & INFORMATION SCIENCES
DEPT OF COMPUTER SCIENCE

CSC311 Computer Algorithms
Second Semester 1428/1429 AH
Final Examination: Tue 20.06.1429 A.H./24.06.2008 C.E. (duration = 3 hours)
Instructor: Dr. Aqil Azmi

---

1. [Marks 20]

Answer *True* or *False*. No need to give reason.

- **a.** If $f(n) = \Theta(g(n)\log n)$ then $g(n) = O(f(n))$.
- **b.** The heapsort sorting algorithm is based on divide and conquer approach.
- **c.** Any problem which can be solved by greedy algorithm can be solved using dynamic programming, while the opposite is not true.
- **d.** Suppose that $f(n)$ is a polynomial of degree k and $g(n)$ is a polynomial of degree ℓ then $f(n) / g(n) = \Theta(n^{k-\ell})$.
- **e.** The adjacency matrix of a directed graph is *never* symmetric.
- **f.** Given two different sorting algorithms both of complexity $O(n \log n)$ then for the same input both will finish simultaneously.
- **g.** In the binary search, if we split the array into two *unequal* parts, *e.g.* $1/5 : 4/5$ then the complexity of the algorithm is $\Theta(\log n)$.
- **h.** Given an ordered list of numbers, we can determine the kth largest element in $O(1)$.
- **i.** Backtracking scheme searches for a solution by going through all the possibilities.
- **j.** In quicksort if we pick the pivot element *randomly* then the worst case running time complexity is $O(n^2)$.

2. [Marks 15]

Mark the *most precise* classification applicable to each of the following pairs of functions. Tick **(A)** if $f(n) = O(g(n))$, but $f(n) \neq \Omega(g(n))$; **(B)** if $f(n) = \Omega(g(n))$, but that $f(n) \neq O(g(n))$; and **(C)** if $f(n) = \Theta(g(n))$.

	$f(n)$	$g(n)$	**A**	**B**	**C**
a.	$2n^2 - 1$	$1000n^2 + 1$			
b.	$\log \sqrt{n-1}$	$\sqrt{1 + \log n}$			
c.	$n + O(\log n)$	$2n$			
d.	$(n+100)^{100}$	$1.01^{\log n}$			
e.	n^n	n^{n+1}			

3. [Marks 15]

Suppose that the running time of algorithm A is given by the recurrence $T(n) = 6T(n / 2) + n^2$. While a competing algorithm B has a running time of $S(n) = aS(n / 3) + n^2$. What is the largest value for a such that B is asymptotically faster than A. Show all your calculations.

4. [Marks 15]

Construct the Huffman tree corresponding to the following set of data:

Letter	a	b	c	d	e	f
Frequency	48	11	10	15	8	4

5. [Marks 20]

Suppose that you are given a sorted sequence of *distinct* integers $\{a_1, a_2, \ldots, a_n\}$. Give an $O(\log n)$ algorithm to determine whether there exists an index i such that $a_i = i$. For example, in {-8, -2, 3, 5, 7}, $a_3 = 3$. While in {2, 3, 4, 5, 6, 7}, there is no such i. Describe and write the algorithm in pseudocode.

6. [Marks 15]

Let the weights $w = \{12, 22, 10, 6\}$ and $m = 28$. Use the *backtracking* algorithm to find all the possible subsets of w that sums to m. Draw the state space tree that corresponds to variable tuple size formulation.

7. [Marks 10 (bonus)]

Show how to multiply the two binary numbers $X = 1100\ 0111$, and $Y = 101\ 1001$ using the divide and conquer based large integer multiplication algorithm. Show all the details.