Multiple-choice Problems

Problem 1. $\sum_{i=1}^{n} i^2 - i$ is

a) $O(n)$ b) $\Theta(n^3)$ c) $\Theta(n^4)$ d) $O(n^2 - n)$ e) $\Theta(n^2)$

Problem 2. Let T(n)=4T($\left\lfloor \frac{n}{2} \right\rfloor$)+2n^2+n and T(0)=1. Then T(n) is

a) Θ(n^2 log n) b) Θ(n^2) c) Θ(n log n)
d) Θ(n log^2 n) e) None

Problem 3. Let T(n)=2T($\lfloor \sqrt{n} \rfloor$)+log(n) and T(0)=1. Then T(n) is

a) Θ(log n) b) Θ(log(n)× log(log(n))) c) Θ(log(log(n)))
d) Θ(log^2 n) e) Θ(n)

Problem 4. How many times do we call the procedure Proc() below?

```
for i:=1 to n do
    for j:=1 to n do
        for k:= 1 to n do
            if i<j<k then
                Proc(i, j, k);
```

a) O(n) b) 1^3+2^3+…+n^3 c) O(n^2) d) O(n^3) e) None

Problem 5. How many comparisons do we need in the worst case if we search for a number x in an arbitrary sequence of n numbers?

a) O(log n) b) O(log^2 n) c) n-1 d) n c) O(n log n)

Problem 6. How many comparisons would the Merge function use in mergesort algorithm in order to merge two sorted arrays (1, 2, 7, 11) (in order) and (3, 4, 5) (in order)?

a) 2 b) 3 c) 4 d) 5 e) 6

Regular Problems:

Problem 7. Consider the following functions:

(func 0) 2^5
(func 1) $4\log(n^3)$
(func 2) 2^n
(func 3) $n + 5$
(func 4) $n!$
(func 5) $\log(n) + 8$
(func 6) n log(n)

Complete the table below. In the entry at row "$func\ i$" and column "$func\ j$", cross the best relation you can prove between "$func\ i$" and "$func\ j$", i.e. put ONLY ONE of O or Ω or Θ. (You do not need to write the proofs).

Functions	$func\ 0$	$func\ 1$	$func\ 2$	$func\ 3$	$func\ 4$	$func\ 5$	$func\ 6$
$func\ 0$	Θ	O	O	O	O	O	O
$func\ 1$	Ω	Θ					
$func\ 2$	Ω		Θ				
$func\ 3$	Ω			Θ			
$func\ 4$	Ω				Θ		
$func\ 5$	Ω					Θ	
$func\ 6$	Ω						Θ

Problem 8. Prove by induction that $\sum_{i=1}^{n} i \times i! = (n+1)! - 1$ for every $n \geq 1$.

Problem 9. The input to this problem is an array A of size n of positive real numbers, and a positive number M. Design an algorithm with running time $O(n)$ to find two indices i and j such that i<j and $\sum_{k=i}^{j} A[k] \leq M$. Your algorithm should find two indices that maximize j-i.

Problem 10. The input to this problem is an array A of size n. Design an algorithm to find the minimum and the second minimum of array A. Your algorithm should use at most n+log(n) comparisons.

Problem 11. Suppose you are choosing between the following three algorithms:

- **a.** Algorithm A solves problems of size n by dividing them into four sub-problems of half the size, recursively solving each sub-problem, and then combining the solutions with cn^2 operations.
- **b.** Algorithm B solves problems of size n by recursively solving two sub-problems of size $n-1$ and then combining the solutions in constant time (c operations).
- **c.** Algorithm C solves problems of size n by dividing them into nine sub-problems of size $n/3$, recursively solving each sub-problem, and then combining the solutions with cn operations.

What are the running times of each of these algorithms (in big-O notation), and which one would you choose?

Problem 12. Explain the heapsort algorithm in detail, write pseudo-code for it, and analyze its running time.