College of Computing and Information Technology

Lecturer: Dr. Nahla Belal

Course: Computing Algorithms (CS 312)

TA: Eng. Mohammad Osama Badawy

Sheet 2

1. Solve the following recurrence relation by expansion:
   a. $T(1) = 1, T(n) = 4\ T(n/3) + n$ for $n > 1$.
   b. $T(1) = 1, T(n) = 2\ T(n/2) + \log n$ for $n > 1$.
   c. $T(1) = 1, T(n) = 2\ T(n/4) + (n)^{1/2}$ for $n > 1$.

2. Write the recurrence relation of the following recursive algorithm. What is the complexity of the algorithm?

```
Test (n)
1       if n = 1 then return 1
2       tmp = test (n/2) + test (n/2)
3       for i = 1 to n do
4               tmp = tmp + i * j
5       end for
6       return tmp
7       end
```

3. Write the recurrence relation of the following recursive algorithm. What is the complexity of the algorithm?

```
Go (n)
1       if n = 1 then return 1
2       return Go (n-1) + Go (n-1)
3       end
```

4. Write the recurrence relation of the following recursive algorithm. What is the complexity of the algorithm?

```
int Test(int n)
int result;
begin
        if (n==1) return (1);
        result = 1;
        for i=1 to n do
                result = result + Test(i-1);
        return (result);
  end
```

5. Given the following recurrence relation:

   $T(1) = 1$, $T(n) = a\,T(n/b) + n^c$ for $n > 1$

   Prove that:

$T(n)$ is $O(n^c)$	if	$a < b^c$
$T(n)$ is $O(n^c \log_b(n))$	if	$a = b^c$
$T(n)$ is $O(n^{\log_b(a)})$	if	$a > b^c$