SLIIT Discover Your Future

BSc (Hons) in Information Technology

Year 2

Data Structures and Algorithms – IT2070

Worksheet 5 – Recursion

2024

1. The power function can be defined as $pow(x,n) = x^n$. This can be evaluated using the multiplication as $x^n = x \times x^{n-1}$ where x is any real number and n is a non-negative integer. [Hint: $pow(x,n-1) = x^{n-1}$]

   a) Write a recursive relation for $pow(x,n)$ where x is any real number and n is a non-negative integer. Clearly define the initial condition(s).

   b) Write a recursive algorithm in pseudo code for the above recursive relation.

   c) Write a recurrence equation that describe the running time T (n) for the above part b) recursive algorithm.

2. Consider a recurrence relation $T(n) = 16T\left(\frac{n}{4}\right) + 10n$. Solve the recurrence relation using the following **Master Theorem** definition.

$$T(n) = \begin{cases} \Theta\left(n^{\log_b a}\right) & f(n) = O\left(n^{\log_b a-\varepsilon}\right) \rightarrow f(n) < n^{\log_b a} \\ \Theta\left(n^{\log_b a} \lg n\right) & f(n) = \Theta\left(n^{\log_b a}\right) \rightarrow f(n) = n^{\log_b a} \\ \Theta(f(n)) & f(n) = \Omega\left(n^{\log_b a+\varepsilon}\right) \rightarrow f(n) > n^{\log_b a} \\ & \text{if } af(n/b) \le cf(n) \text{ for c} < 1 \text{ and large } n \end{cases}$$

3. Consider the function $f(n)$, which is defined below. n is a nonnegative integer.

$$f(n) = \begin{cases} n/4 & n \text{ is even} \\ f(n+1) & n \text{ is odd} \end{cases}$$

a) Use the above equation to manually compute $f(3)$.

b) Identify the base and recursive component of the function definition.

c) Write a recursive algorithm in pseudo code for the above recursive relation $f(n)$.

4. The function $sum(n)$ is defined as the sum of integers from 1 to n.

$sum(n) = 1 + 2 + 3 + 4 \,......... + n$

a. Write a recursive relation for $sum(n)$ where n is a non-negative integer. Clearly define the initial condition(s). [Hint: $sum(n-1) = 1 + 2 + 3 + 4 \,......... + (n-1)$]

b. Write a **recursive** and **iterative** algorithms in pseudo code for the above recursive relation.

c. Write a recurrence equation that describe the running time T (n) for the above part b) **recursive** algorithm.