National University of Computer and Emerging Sciences, Lahore Campus

Course:	Design and Analysis of Algorithms	Course Code:	CS-2009
Program:	BS (CS)	Semester:	Spring 2025
Due Date:	Feb 18, 2025 (by 4:10 PM in class)	Total Marks:	80
Sections:	CS-6A		
Exam:	Assignment#1		

Instruction/Notes:

Instructions: Plagiarism in any form will not be tolerated. Only handwritten, hard copy assignments will be accepted.

Q#1: Derive the recurrence of following recursive algorithms. (Marks 12)

Part(a)

```
Split Data (Arr[], left, right){
  If (left < right) {
    Split Data (Arr, left, right-1)
    Split Data (Arr, left+1, right)
    View Data (Arr, left, right)
  }
}
View Data (Arr[], left, right){
  i=left
  N_Arr[left+right]
  While (i < right) {
    j = left
    k = 1
    while (j < right) {
      N_Arr[k] = Arr[j]
      print Arr[i]
      k = k+1
      j = j*2
    }
    i++
  }
}
```

Part(b)

```
Temp(A[], B[], N){
  if(N==1)
    return
  Temp(A+N/2,B, N/2) + Temp(A, B + N/2, N/2)
  Temp(A+N/2,B, N/2) + Temp(A, B + N/2, N/2)
  Temp(A+N/2,B, N/2) + Temp(A, B + N/2, N/2)
  Temp(A+N/2,B, N/2) + Temp(A, B + N/2, N/2)
  Temp(A+N/2,B, N/2) + Temp(A, B + N/2, N/2)
  Temp(A+N/2,B, N/2) + Temp(A, B + N/2, N/2)
  Temp(A+N/2,B, N/2) + Temp(A, B + N/2, N/2)
  Temp(A+N/2,B, N/2) + Temp(A, B + N/2, N/2)
  Solve(A,B,N)
}
Solve (A[], B[], N){
  for(i=1 to N){
    for(j=1 to i){
      print A[i]*B[j]
    }
  }
}
```

Part (c):

```
Mystery (N){
  If (N > 1){
    Print "Deriving recurrence is fun"
    Mystery (2N⁄3)
    for (i=1 to N)
      Print "Solving recurrence is fun"
    Mystery (N⁄5)
  }
}
```

Part(d):

```
Stooge-Sort (Arr [], left, right){
  if (Arr[left] > A[right])
    exchange(A[left], A[right])
  if (left > right)
    return
  else{
    third = (right - left + 1)/3
    Stooge-Sort (Arr, left, right - third)
    Stooge-Sort (Arr, left + third, right)
    Stooge-Sort (Arr, left, right - third)
  }
}
```

Also provide the reason that how these recursive calls will sort the array and why we need the third recursive call with data ranging from Left to 3rd quarter of the array.

Q#2: Solving Recurrence (Marks: 16*4=64)

Use a recursion tree method to determine a good asymptotic upper bound on following recurrences. Please refer to the Appendix of your textbook for using arithmetic, geometric and harmonic series. Clearly show your working and don't forget to mention the relevant formula before writing the final answer.

Part(a): Linear Decay

a) $T(N) = 3T(N-1) + \Theta(1)$
b) $T(N) = T(N-2) + O(N^2)$
c) $T(N) = 2T(N-1) + \Theta(N)$
d) $T(N) = T(N-1) + O(\frac{1}{N})$

Part(b): Exponential decay with balanced split of data (Attempt any four questions for assignment from part b)

e) $T(N) = 2T\left(\frac{N}{4}\right) + O(logN)$
f) $T(N) = 10T\left(\frac{N}{2}\right) + \Theta(1)$
g) $T(N) = 2T\left(\frac{N}{2}\right) + O(\frac{N}{logN})$
h) $T(N) = 4T\left(\frac{N}{2}\right) + O(N^2\sqrt{N})$
i) $T(N) = 2T\left(\frac{N}{4}\right) + O(\sqrt{N})$
j) $T(N) = 3T\left(\frac{N}{2}\right) + O(N^3)$
k) $T(N) = 7T\left(\frac{N}{5}\right) + \Theta(1)$
l) $T(N) = 3T\left(\frac{2N}{3}\right) + O(1)$
m) $T(N) = 3T\left(\frac{4N}{5}\right) + O(1)$

Part(c): Exponential decay with unbalanced split of data

n) $T(N) = T\left(\frac{N}{4}\right) + T\left(\frac{3N}{4}\right) + O(N^2)$
o) $T(N) = T\left(\frac{N}{10}\right) + T\left(\frac{9N}{10}\right) + O(N)$
p) $T(N) = T\left(\frac{2N}{7}\right) + T\left(\frac{5N}{7}\right) + O(N^2)$

Q#3: Asymptotic Analysis (4 Marks)

a) **Prove that $T(N) = \Theta(N^3)$ by finding appropriate constants.**

$$T(N) = \frac{1}{8}N^3 - 5N^2$$

b) **Prove that $T(N) = \Theta(g(N))$ by finding the appropriate g(N) and constants.**

$T(N) = 5\sqrt{N} + 3N^2 logN + \frac{N}{(logN)^2}$	$T(N) = N^{10} + 2^{3N} + 5^{N+4} + 100N logN$