



HABIB UNIVERSITY

Data Structures & Algorithms

CS/CE 102/171 Spring 2023

Instructor: Maria Samad

Recurrence Equations

Student 1: _____

A. For the given recursive functions, derive their respective Recurrence Equations:

1. def recursive_function_1(n): if (n == 0): return 1 else: return recursive_function_1(n - 10)	$T(n) = \begin{cases} 1, & n=0 \\ T(n-10)+1, & n>0 \end{cases}$
2. def recursive_function_2(n): if (n == 0): return 1 else: for i in range(n): for j in range(n): print(i, j) return recursive_function_2(n - 5)	$T(n) = \begin{cases} 1, & n=0 \\ T(n-5)+n^2+1, & n>0 \end{cases}$
3. def recursive_function_3(n): if (n == 0): return 1 else: i = 1 while (i*i < n): i += 1 return recursive_function_3(n - 2)	$T(n) = \begin{cases} 1, & n=0 \\ T(n-2)+\sqrt{n}+1, & n>0 \end{cases}$
4. def recursive_function_4(n): if (n == 0): return 1 else: a = 1 while (a < n): a = a * 3 return recursive_function_4(sqrt(n))	$T(n) = \begin{cases} 1, & n=0 \\ T(\sqrt{n})+\log_3 n+1, & n>0 \end{cases}$
5. def recursive_function_5(n): if (n == 0): print(1) else: for i in range(n): j = 1 while (j < n): j = j * 2 recursive_function_5(n - 2) recursive_function_5(n - 3)	$T(n) = \begin{cases} 1, & n=0 \\ T(n-2)+T(n-3)+n\log n+1, & n>0 \end{cases}$

i runs n times
j runs log n times
 $\therefore i \times j = n \log n$
times

i = 0
j = log n times

$i=0 \quad i=1 \quad i=2 \quad \dots \quad i=n$
 $j \times \quad j=0 \text{ (1 time)} \quad j=0, 1 \text{ (2 times)} \quad j=n \text{ times}$
 $1 \times 0 + 1 \times 1 + 1 \times 2 + \dots + 1 \times n = 1 + 2 + 3 + \dots + n$
 $O(n^2) \quad \sim \frac{n(n+1)}{2}$

6. def recursive_function_6(n): $\sim T(n)$
 if (n == 0): $\{ 1$
 return 1
 else:
 for i in range(n): $\{ n^2 \text{ times}$
 for j in range(i):
 print(i, j)
 return recursive_function_6(n - 5) $\sim T(n-5)$
 return recursive_function_6(n - 5) $\sim T(n-5)$
 return recursive_function_6(n - 5) $\sim T(n-5)$

For $n > 0$:
 $T(n) = T(n-5) + T(n-5) + T(n-5) + n^2 + 1$
 $\Rightarrow T(n) = 3T(n-5) + n^2 + 1$
 $\therefore T(n) = \begin{cases} 1 & n=0 \\ 3T(n-5) + n^2 + 1, & n > 0 \end{cases}$

7. def recursive_function_7(n): $\sim T(n)$
 if (n == 0): $\{ 1$
 return 1
 else:
 recursive_function_7(n//3) $\sim T(n/3)$
 for i in range(n): $\{ n$
 print(i)
 for j in range(n): $\{ n$
 print(j)

For $n > 0$:
 $T(n) = T(n/3) + n + n + 1$
 $= T(n/3) + 2n + 1$
 $\therefore T(n) = \begin{cases} 1 & n=0 \\ T(n/3) + 2n + 1, & n > 0 \end{cases}$

8. def recursive_function_8(n): $\sim T(n)$
 if (n == 0): $\{ 1$
 return 1
 else:
 recursive_function_8(n//2) $\sim T(n/2)$
 recursive_function_8(n//3) $\sim T(n/3)$

$T(n) = \begin{cases} 1 & n=0 \\ T(n/2) + T(n/3) + 1, & n > 0 \end{cases}$

9. def recursive_function_9(n): $\sim T(n)$
 if (n < 2): $\{ 1$
 return 7
 else:
 a = 1 $\{ \log_7 n$
 while (a < n):
 a = a // 7
 recursive_function_9(n - 3) $\sim T(n-3)$
 recursive_function_9(n/3 + 1) $\sim T(n/3 + 1)$
 recursive_function_9($\sqrt{n} - 5$) $\sim T(\sqrt{n} - 5)$

For $n \geq 2$:
 $T(n) = 1 + \log_7 n + T(n-3) + T(n/3 + 1) + T(\sqrt{n} - 5)$
 $\therefore T(n) = \begin{cases} 1 & n < 2 \\ T(n-3) + T(n/3 + 1) + T(\sqrt{n} - 5) + \log_7 n, & n \geq 2 \end{cases}$

10. def recursive_function_10(n): $\sim T(n)$
 if (n < 5): $\{ 1$
 return -1
 else:
 for i in range(n): $\{ n \text{ times}$
 recursive_function_10(n/2) $\sim T(n/2)$

For $n \geq 5$:
 $T(n) = n * T(n/2) + 1$
 $\therefore T(n) = \begin{cases} 1 & n < 5 \\ n * T(n/2) + 1, & n \geq 5 \end{cases}$

$\therefore n * T(n/2)$