**Data Structures - Fall 2019**

**Practice Exam 1 (v2)**

s0\_name\_and\_id.py

**Open the file and make sure you type your last name, first name, and UTEP ID**

s1\_recursion.py

**Problem 1 (10 points):** Provide and implementation for the *count* method. This method receives a non-negative int n and returns the number of digits in n that are less than or equal

to 5 - Use recursion - no loops.

Example1: count(285) -> 2

Example2: count(565891) -> 3

s2\_iterative\_time\_complexity.py

**Problem 2 (4 points):** What is the running time (big-O) of the following function?

**def** f(a):

**for** i **in** range(len(a)):

print(a[0])

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Option 0**  O(1) | **Option 1**  O(n) | **Option 2**  O( log(n) ) | **Option 3**  O(n ^ 2) | **Option 4**  O( n log(n) ) | **Option 5**  O( n ^ 3) |

**Problem 3 (4 points):** What is the running time (big-O) of the following function?

**def** f(a):

**for** i **in** range(256):

print(a[0])

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Option 0**  O(1) | **Option 1**  O(n) | **Option 2**  O( log(n) ) | **Option 3**  O(n ^ 2) | **Option 4**  O( n log(n) ) | **Option 5**  O( n ^ 3) |

**Problem 4 (4 points):** What is the running time (big-O) of the following function?

**def** f(a):

**for** i **in** range( len(a) // 2 ):

**for** j **in** range( len(a) // 4 ):

**for** k **in** range( len(a) // 8):

print(a[0])

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Option 0**  O(1) | **Option 1**  O(n) | **Option 2**  O( log(n) ) | **Option 3**  O(n ^ 2) | **Option 4**  O( n log(n) ) | **Option 5**  O( n ^ 3) |

**Problem 5 (4 points):** What is the running time (big-O) of the following function?

**def f(a):**

**for k in range(len(a)):**

**i = len(a)**

**while i > 0:**

**print(a[0])**

**i = i // 2**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Option 0**  O(1) | **Option 1**  O(n) | **Option 2**  O( log(n) ) | **Option 3**  O(n ^ 2) | **Option 4**  O( n log(n) ) | **Option 5**  O( n ^ 3) |

s3\_recursive\_time\_complexity.py

**Problem 6 (4 points):** What is the recurrence equation that describes the running time of the following recursive function? T(n) = a T(n / b) + n ^ k. What are the values of a, b, and k?

**def f(a, n): *# First call: f(a, len(a))***

**if n > 0:**

**f(a, n // 2)**

**f(a, n // 2)**

**for i in range(len(a)):**

**print(a[0])**

**Problem 7 (4 points):** What is the recurrence equation that describes the running time of the following recursive function? T(n) = a T(n / b) + n ^ k. What are the values of a, b, and k?

**def f(a, n): *# First call: f(a, len(a))***

**if n > 0:**

**for i in range(2):**

**f(a, n // 2)**

**f(a, n // 2)**

**for i in range(len(a) \* len(a)):**

**print(a[0])**

**Problem 8 (4 points):** What is the recurrence equation that describes the running time of the following recursive function? T(n) = a T(n / b) + n ^ k. What are the values of a, b, and k?

**def f(a, n): *# First call: f(a, len(a))***

**if n > 0:**

**f(a, n // 4)**

**f(a, n // 4)**

**f(a, n // 4)**

**for i in range(10):**

**print(a[0])**

**------------------------------------------------------------------------------------------------------------------------------**

**Master Theorem:**

T(n) = O() if a>

T(n) = O(log n) if a=

T(n) = O() if a<

**Problem 9 (4 points):** Solve the following recurrence equation:

T(1) = 1

T(n) = 8 T( n / 2) +

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Option 0**  O() | **Option 1**  O() | **Option 2**  O( log n) | **Option 3**  O(log n) | **Option 4**  O( n log(n) ) | **Option 5**  O( n ) |

**Problem 10 (4 points):** Solve the following recurrence equation:

T(1) = 1

T(n) = 8 T( n / 2) + 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Option 0**  O() | **Option 1**  O() | **Option 2**  O( log n) | **Option 3**  O(log n) | **Option 4**  O( n log(n) ) | **Option 5**  O( n ) |

**Problem 11 (4 points):** Solve the following recurrence equation:

T(1) = 1

T(n) = 2 T( n / 2) +

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Option 0**  O() | **Option 1**  O() | **Option 2**  O( log n) | **Option 3**  O(log n) | **Option 4**  O( n log(n) ) | **Option 5**  O( n ) |

**Problem 12 (4 points):** Consider the following recurrence equation: T(n) = 2 T (n / 2) + 1

Solve the equation by iteration.

T(1) = 1

T(2) = ?

T(4) = ?

T(8) = ?

T(16) = ?

T(32) = ?

What is T(32) ? Your answer must be an integer

s4\_activation\_records.py

To answer the following three questions, trace the execution of p1(2) using activation records. Every time you create an activation record, assign it an ID starting from 0.

**def** p1(n):

**if** n > 0:

p2 (n - 1)

print(**"p1, n = "**, n )

**def** p2(n):

**if** n > 0:

p2 (n // 2)

p1 (n - 1)

print(**"p2, n = "**, n )

**Problem 13 (5 points):** How many activation records did you create? The initial call to p1 counts as the first activation record. If you drew 3 boxes, the answer should be 3. If you drew 5 boxes, the answer should be 5, etc.

**Problem 14 (5 points):** What is the value of n in activation record #2? Activation record #0 is the one you created for p1(2).

**Problem 15 (5 points):** What is the FIRST line printed on the console?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Option 0**  p1, n = 0 | **Option 1**  p2, n = 1 | **Option 2**  p1, n = -1 | **Option 3**  p1, n = 2 | **Option 4**  p2, n = 2 | **Option 5**  p2, n = 0 |

s5\_lists\_1.py

To answer the following three questions, trace the following piece of code:

***class*** *Node:*

***def*** *\_\_init\_\_(self, item, next):*

*self.item = item*

*self.next = next*

*x =* ***None***

*y =* ***None***

***for*** *i* ***in*** *range(2):*

*x = Node(i + 2,y)*

*y = Node(i + 1,x)*

*print(y.item) # Print statement 0*

*x.next.item += 2*

*y.next.next.item += 4*

*print(y.next.item) # Print statement 1*

*y=x.next.next*

*print(y.item) # Print statement 2*

**Problem 16 (5 points):** What integer does 'Print statement 0' print to the console?

**Problem 17 (5 points):** What integer does 'Print statement 1' print to the console?

**Problem 18 (5 points):** What integer does 'Print statement 2' print to the console?

s6\_lists\_2.py

**Problem 19 (7 points):** Complete the implementation of the method *add\_last* <- Method that receives an item as input, creates a node that stores this value, and appends it to the end of this list

**Problem 20 (7 points):** Complete the implementation of the method *index\_of* <- Method that receives an item as input and returns the index of the node that stores item. If item is not in the list, you should return -1

**Problem 21 (7 points):** Complete the implementation of the method *get\_last* <- Method that returns the item associated with the last node in the list. If no such node exists, return None

**Problem 22 (7 points):** Complete the implementation of the method *size* <- Method that returns the number of nodes in the list

Total number of points: 112 [---12 points of extra credit---]