

Part 1 Answer True/False and Multiple Choice questions

1. Which of the following is a $\Theta(n)$ operation?

- (a) Sorting a list with Selection sort
- (b) Finding the i^{th} item in a Python list.
- (c) Re-assigning the element at the end of a Python list.
- (d) Deleting an item from the middle of a Python list.

2. Which of the following is **not** true of Python dictionaries?

- (a) They are implemented as hash tables.
- (b) Lookup is very efficient.
- (c) Values must be immutable.
- (d) All of the above are true.

3. How many iterations will the while loop of the *Binary Search* do when searching for 21 in the sequence [1, 5, 12, 14, 17, 21, 28]? Use the Binary Search algorithm I presented in class.

- (a) 5
- (b) 4
- (c) 3
- (d) 2

Part 2. Answer short-answer questions

1. Consider the following code fragment:

```
from ListNode import *

z = ListNode(34)
y = ListNode(25,z)
x = ListNode(12,y)
t = ListNode(20,y)
```

What will be produced by this code fragment (draw a pictorial representation)?

For your reference, the definition of the `ListNode` class:

```
class ListNode:
    def __init__(self, item = None, link = None):
        '''creates a ListNode with the specified data value and link
link'''
        self.item = item
        self.link = link
```

2. Give a theta analysis of the time efficiency of the following code fragment. Provide explanations.

```
n = int(input("Enter a positive integer:"))
myList = []
while n > 1:
    myList.insert(0, n)
    n -= 3
```

$T(n) = \Theta(\quad)$

3. Give pictorial representation of the Python's memory during execution of the code given below. Show the result of print statements.

```
def func(a,b,c):
    a.append(c)
    b = b + ", world!"
    c = c/5
    a = [1,2,3]
    print(a,b,c)
```

```
def main():
    l = ['a','b']
    d = "Hello"
    k = 25
    func(l,d,k)
    print(l,d,k)
```

Part 3. Coding and related to coding questions

Python has a `set` type that efficiently implements mathematical sets. You can get information on this container class by consulting reference documents or typing `help(set)` at a Python prompt.

Suppose you are implementing your own `Set` class that includes `add`, `remove`, `clear`, `__contains__`, `intersection`, `union`, and `difference` operations.

Utilizing each of the following concrete data structures, explain how you would implement the required operations and provide an analysis of the run-time efficiency of each operation.

(a) an unordered Python list.

(b) a sorted Python list.

(c) a Python dictionary. (Note: the elements of the set will be the keys, you can just use `None` or `True` as the value.)

Use any of the data structures to implement your version of the `Set` class.