

BRONX COMMUNITY COLLEGE
of The City University of New York
DEPARTMENT OF MATHEMATICS and COMPUTER SCIENCE
CSI 33
Final Exam Sample/outline Solutions

1 Part 1

Do all problems in this part. You are to answer True/False and Multiple Choice questions (10-11 questions)

1. Python lists are implemented using contiguous arrays: **True**
2. Inserting into the middle of array-based implementation of list is a $\Theta(n)$ operation: **True**
3. Looking up an item in a Python dictionary, given a key, is a $\Theta(n)$ operation: **False**
4. Which of the following is **not true** of Python list (choose only one)?
 - (a) They are implemented underneath as contiguous arrays
 - (b) They allow for efficient random access
 - (c) They can grow and shrink dynamically
 - (d) **All items in a list must be of the same type**
5. What operation is not supported for Python dictionaries?
 - (a) **Item ordering (sorting)**
 - (b) Item insertion
 - (c) Item deletion
 - (d) Item lookup
6. When using a linked implementation of a queue, where should insertions be done?
 - (a) at the front (head) of the linked list
 - (b) in the middle of the list
 - (c) **at the end (tail) of the list**

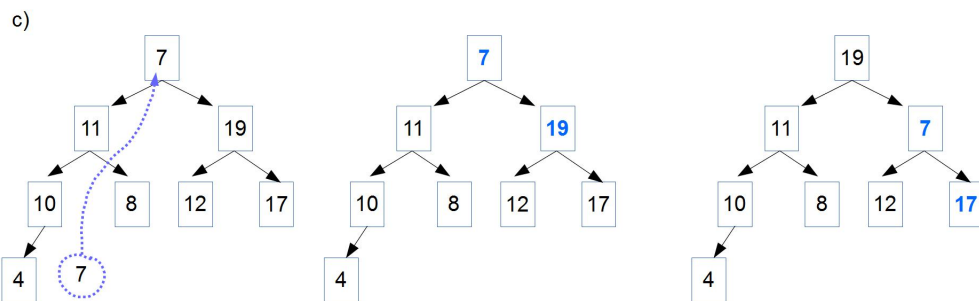
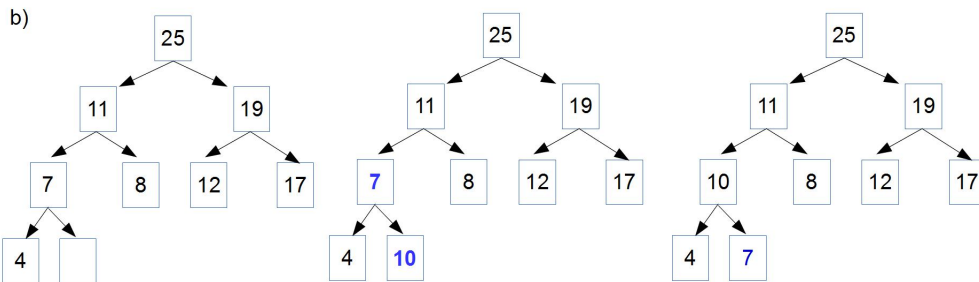
- (d) anywhere in the list, just keep the information of its location
7. A C++ function must return a value: **False**
8. The *scope* of a variable refers to
- (a) the different values it can hold
 - (b) **the section of code where the variable can be accessed**
 - (c) the time during which memory is allocated for the variable
 - (d) the name of the variable.

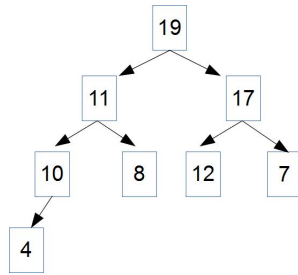
2 Part 2

Answer all questions in this part (4-5 questions).

1.

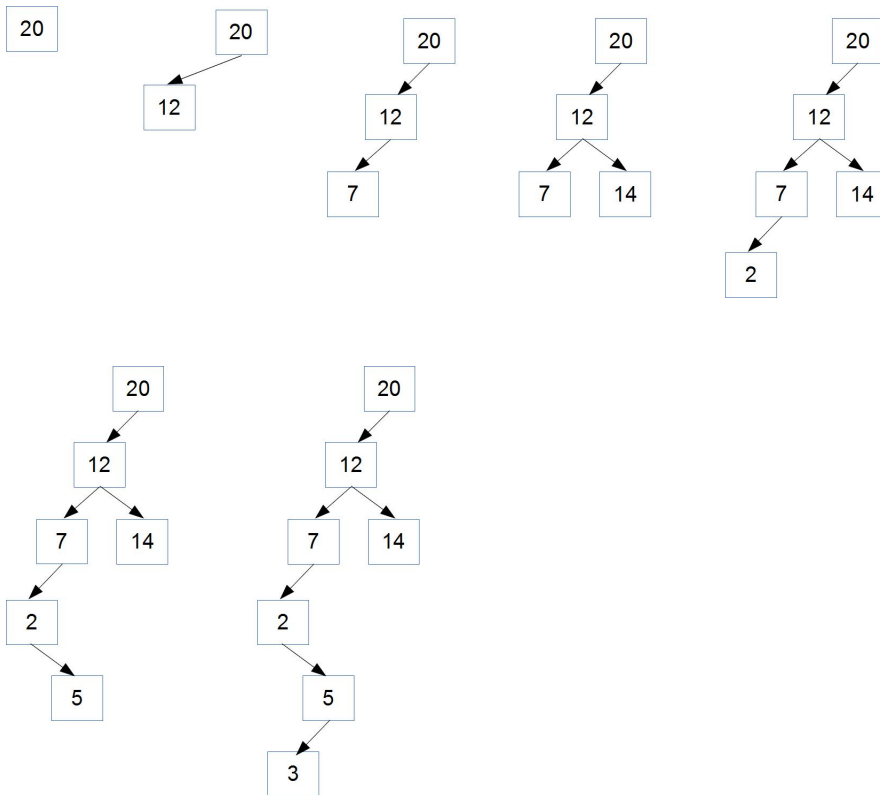
(a) [*None*, 25, 11, 19, 7, 8, 12, 17, 4]





2. 20, 14, 12, 8, 7, 5, 3, 2

3.



4. The adjacency matrix:

	A	B	C	D	E
A	0	1	1	0	0
B	0	0	0	1	0
C	0	1	0	0	0
D	0	0	0	0	1
E	0	0	0	0	0

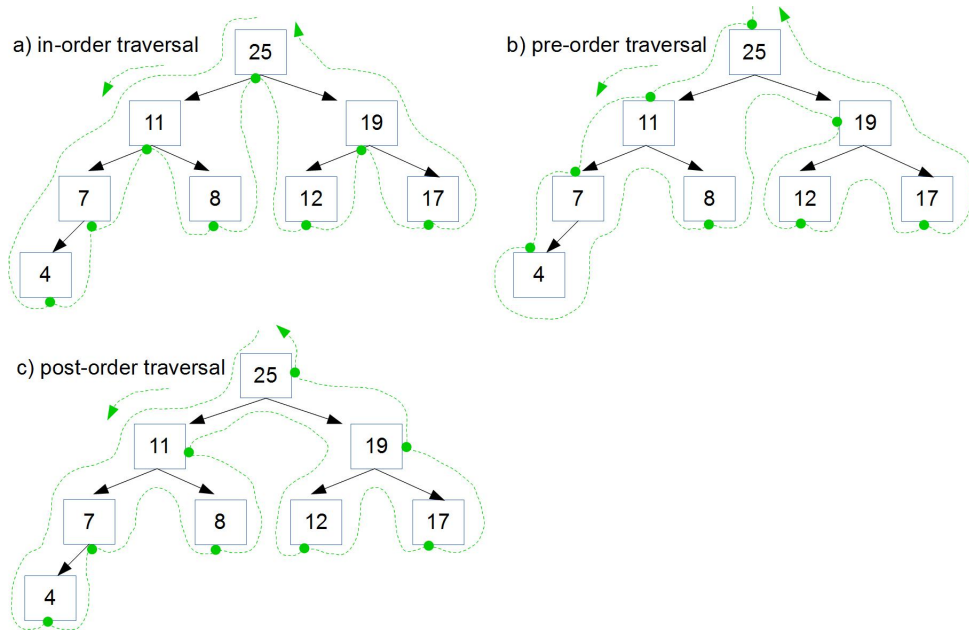
Using Python list:

```
[
[0 , 1 , 1 , 0 , 0],
[0 , 0 , 0 , 1 , 0],
[0 , 1 , 0 , 0 , 0],
[0 , 0 , 0 , 0 , 1],
[0 , 0 , 0 , 0 , 0]
]
```

The adjacency list (without weights):

```
[
['A', ['B', 'C']],
['B', ['D']],
['C', ['B']],
['D', ['E']],
['E', []]
]
```

5.



inorder: 4, 7, 11, 8, 25, 12, 19, 17

preorder: 25, 11, 7, 4, 8, 19, 12, 17

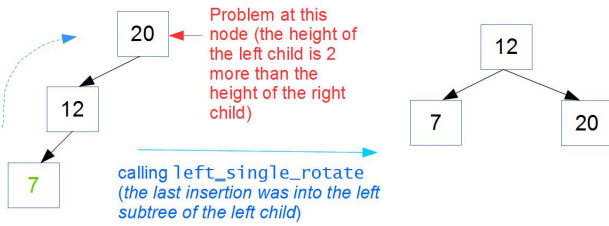
postorder: 4, 7, 8, 11, 12, 17, 19, 25

3 Part 3

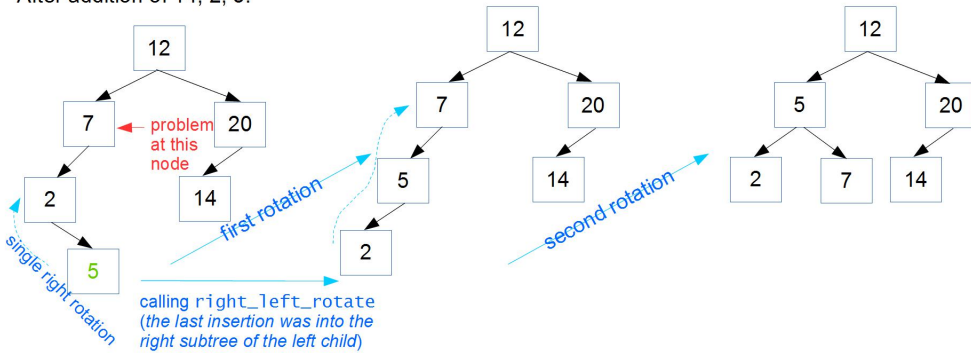
Do any 3 problems out of 5.

1.

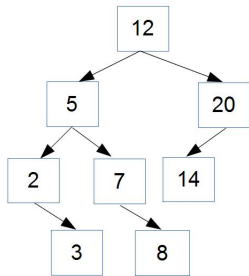
After addition of 20, 12, 7:



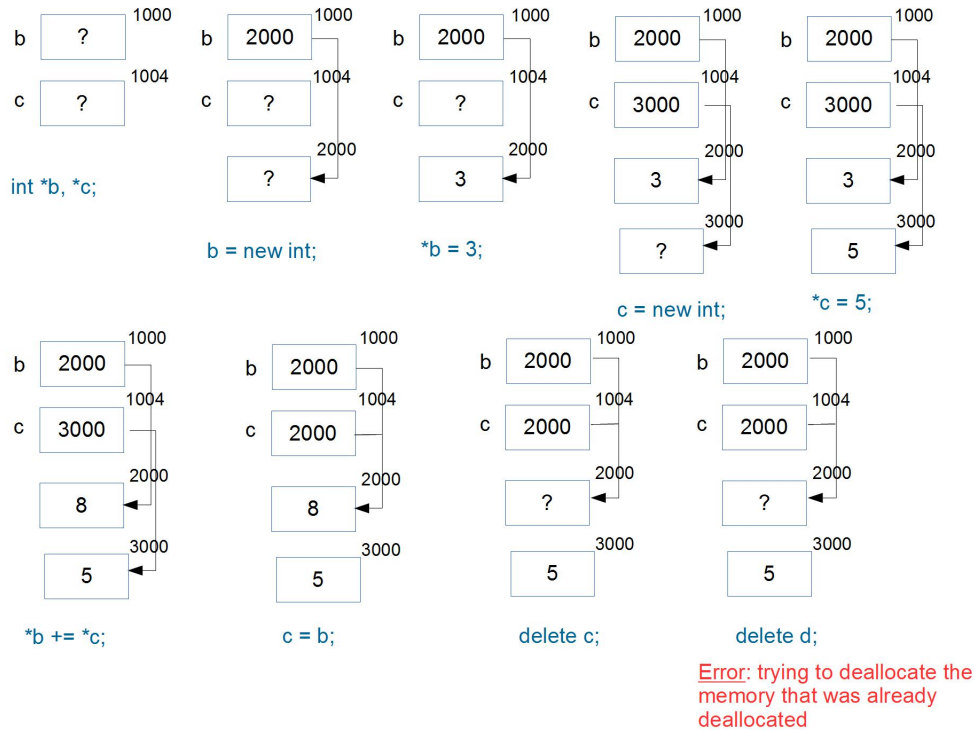
After addition of 14, 2, 5:



After addition of 3, 8:



2. We have a memory leak (lost information about the memory allocated for integer, with 5 in it).



One way to fix it is to remove line `c=b;`, another is to use temporary storage `tmp`, before the assignment `c=b;`:

```
tmp = c;
c = b;
delete c;
delete tmp;
```

3. posted separately as program

4. Let's use hash function to get the locations for each of the value to store

$f(4005) = 4005\%7 = 1$, - put value 4005 to the memory location 1.

$f(1908) = 1908\%7 = 4$, - put value 1908 to the memory location 4.

$f(7890) = 7890\%7 = 1$, - put value 7890 to the memory location 1. Since the slot is already occupied, by separate chaining approach, create a node and link it to the node at location 1.

$f(1928) = 1928\%7 = 3$, - put value 1928 to the memory location 3.

$f(0035) = 0035\%7 = 0$, - put value 0035 to the memory location 0.

$f(1076) = 1076\%7 = 5$, - put value 1076 to the memory location 5.

$f(0187) = 0187\%7 = 5$, - put value 0187 to the memory location 5. Since the slot is already occupied, by separate chaining approach, create a node and link it to the node at location 5.

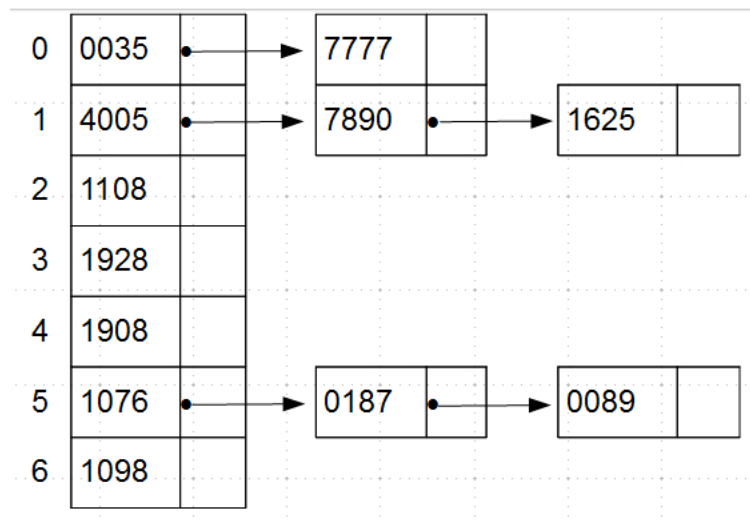
$f(1098) = 1098\%7 = 6$, - put value 1098 to the memory location 6.

$f(7777) = 1928\%7 = 0$, - put value 7777 to the memory location 0. Since the slot is already occupied, by separate chaining approach, create a node and link it to the node at location 0.

$f(1108) = 1108\%7 = 2$, - put value 1108 to the memory location 2.

$f(0089) = 0089\%7 = 5$, - put value 0089 to the memory location 0. the slot is already occupied, hence create a node and link it to the previous node at location 5 (node with 0187).

$f(1625) = 1625\%7 = 1$, - put value 1625 to the memory location 1. the slot is already occupied, hence create a node and link it to the previous node at location 0 (node with 7890).



5. Solution is posted in a separate pdf file.