COM2067 DATA STRUCTURES

EXERCISE QUESTIONS FOR FINAL EXAM

1. Using the two different methods below, insert the given values to the hash table that can contain at most 10 elements. For this question, simply fill in the table below.

(a) Linear Probing

$h(k) = k \bmod 10$

(b) Quadratic Probing

$h(k, i) = [h'(k) + c_1 i + c_2 i^2] \bmod 10$

where $h' = k \bmod 10$, $c_1 = 3$, $c_2 = 1$

35, 46, 45, 50, 26, 12, 55, 17, 7

(a)

0	1	2	3	4	5	6	7	8	9

(b)

0	1	2	3	4	5	6	7	8	9

Solution for (b):

0	1	2	3	4	5	6	7	8	9
50	7	12	55	26	35	46	17		45

1) $h(35, \overset{i=0,\ \text{initially.}}{0}) = [\underbrace{h'(35)}_{35//10=5} + 3 \cdot 0 + 1 \cdot 0] // 10 = 5$

2) $h(46, 0) = [\underbrace{h'(46)}_{6} + 3 \cdot 0 + 1 \cdot 0] // 10 = 6$

3) $h(45, 0) = [5 + \overset{3 \cdot 0}{0} + \overset{1 \cdot 0}{0}] // 10 = 5$ is occupied.

$h(45, \underset{i=1,\ \text{now.}}{1}) = [\underbrace{5 + 3 \cdot 1 + 1 \cdot 1^2}_{9}] // 10 = 9$

4) $h(50, 0) = [0 + 0 + 0] // 10 = 0$

5) $h(26, 0) = [6 + 0 + 0] // 10 = 6$ is occupied

$h(26, 1) = [\underbrace{6 + 3 \cdot 1 + 1 \cdot 1^2}_{10}] // 10 = 0$ is also occupied.

$h(26, \underset{i=2,\ \text{now.}}{2}) = [\underbrace{6 + 3 \cdot 2 + 1 \cdot 2^2}_{14}] // 10 = 4$

6) $h(12, 0) = [12 + 0 + 0] // 10 = 2$

7) $h(55, 0) = [\underbrace{5 + 0 + 0}_{5}] // 10 = 5$ is occupied.

$h(55, 1) = [\underbrace{5 + 3 + 1}_{9}] // 10 = 9$ is also occupied

$h(55, 2) = [\underbrace{5 + 3 \cdot 2 + 1 \cdot 4}_{15}] // 10 = 5$ is occupied.

$h(55, 3) = [\underbrace{5 + 3 \cdot 3 + 1 \cdot 9}_{23}] // 10 = 3$

8) $h(17, 0) = [7 + 0 + 0] // 10 = 7$

9) $h(7, 0) = [7 + 0 + 0] // 10 = 7$ is occupied.

$h(7, 1) = [\underbrace{7 + 3 + 1}_{11}] // 10 = 1$

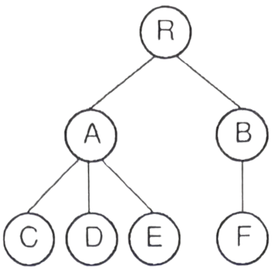
Solution for (a):

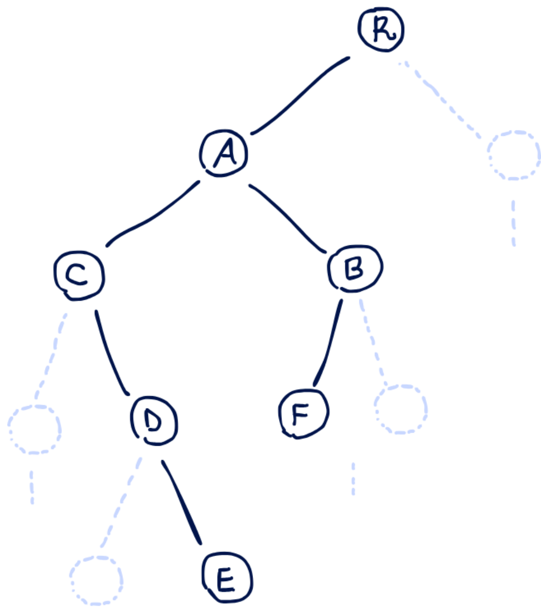
0	1	2	3	4	5	6	7	8	9
50	17	12	7		35	46	45	26	55

1) h(35) = 35 // 10 = 5th cell
2) h(46) = 46 // 10 = 6th cell
3) h(45) = 5, 5th cell is occupied. next available cell is 7.
4) h(50) = 0th cell.
5) h(26) = 6th cell is occupied. next available cell is 8
6) h(12) = 2nd cell.
7) h(55) = 5, 5th cell is occupied next available cell is 9.
8) h(17) = 7th cell is occupied. next available cell is 1.
9) h(7) = 7th cell is occupied. next available cell is 3.

---

2. Convert the given general tree into a binary tree.c





3. Write the post-order traversal of the created binary.

post-order traversal: **E D C F B A R**

---

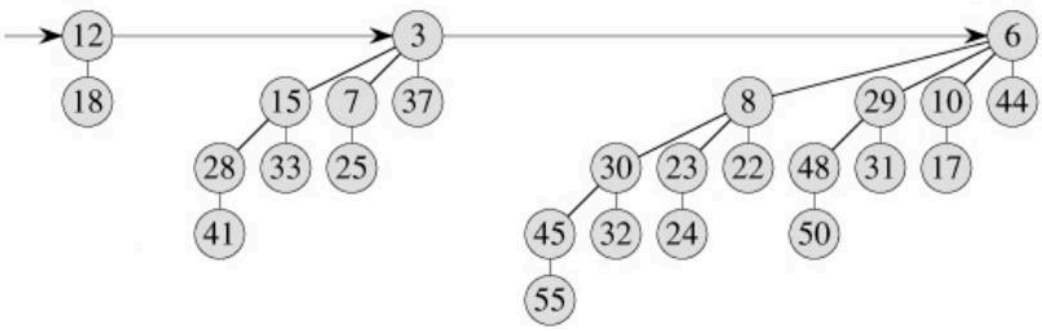
4. Pre-order and in-order traversals of a binary tree are given as follows:

Pre-order: 8, 5, 9, 7, 1, 12, 2, 4, 11, 3

Post-order: 9, 5, 1, 7, 2, 12, 8, 4, 3, 11

---

5. Extract the key with value 48 from the following binomial heap. Draw the produced binomial heap after this operation. Show all of the steps of producing this binomial heap.



6. Insert the following integer keys into a red-black tree. Show the red-black tree after each insertion that causes change in the tree.

24, 18, 43, 57, 90, 35, 60

---

7. Construct a binary tree from the following traversal results. Describe the steps.

In-order Traversal: 6 4 1 3 0 5 9 7 8 2

Pre-order Traversal: 0 1 4 6 3 2 5 7 9 8

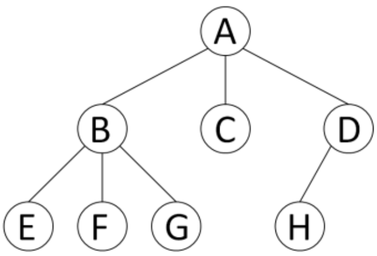
---

8. Insert the following integer keys into an AVL tree. After each balancing (rotation) operation, write which rotation you have performed and show the AVL tree.

50, 40, 25, 10, 15, 5, 45, 35, 20, 30

---

9. Convert the given general tree into a binary tree. Show each step.



---

10. Apply the following insert and delete operations on 2-3 tree. Show 2-3 tree after each operation.

Insert 25, 15, 5, 30, 20, 35, 40

Delete 30

---

11. Create a Huffman tree using the following uppercase letters. Assume that only those letters are used in the text. Given values represent the estimated frequencies of each letter in a paragraph. Using the created Huffman tree, specify a binary coding for each letter.

P	S	M	Y	K	E	A
4	6	8	9	12	14	16

12. Insert the following integer keys into a red-black tree. Show the red-black tree after each insertion that causes change in the tree.

25, 46, 51, 16, 20, 75, 65

---

13. Insert the following integer keys into an AVL tree. Show the AVL tree after each insertion.

65, 75, 20, 16, 51, 46, 25

Then, delete keys 16 and 46 from the AVL tree. Show the AVL tree after each deletion.

---

14. Construct a binary tree from the following traversal results. Describe the steps.

In-Order Traversal: G E B D A F J H I C

Pre-Order Traversal: A B E G D C F H J I

---

15. Apply the following insert and delete operations on B tree (m = 5). Show B tree after each operation.

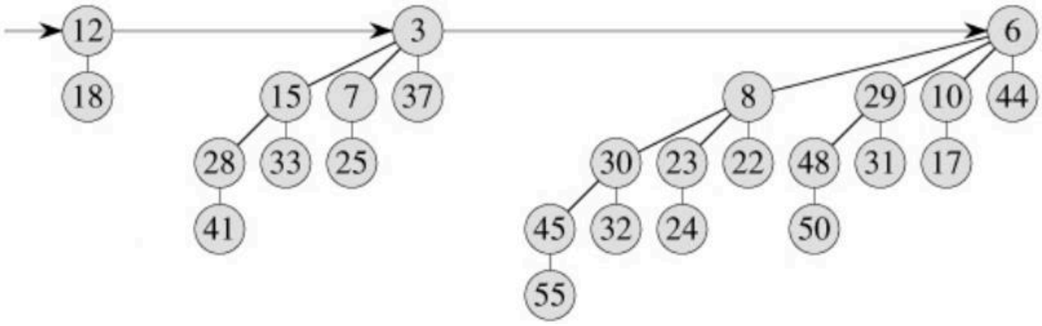
Insert 75, 20, 16, 51, 46, 25, 49, 59, 81, 33, 23, 50

Delete 16, 46

Insert 37, 27

---

16. Extract the node with minimum key from the following binomial heap. Draw the produced binomial heap after this operation.



---

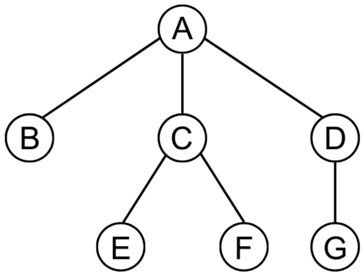
17. Apply the following insert and delete operations on binary search tree (BST). Show the binary search tree after each operation.

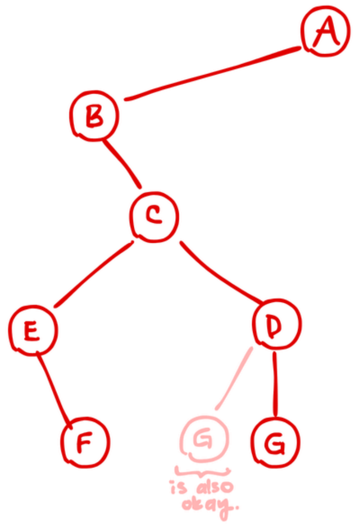
Insert 35, 38, 65, 41, 28

Delete 35, 28

Insert 39, 16, 81

18. Convert the given general tree into a binary tree.





19. Create a Huffman tree using the following uppercase letters. Assume that only those letters are used in the text. Given values represent the estimated frequencies of each letter in a paragraph. Using the created Huffman tree, specify a binary coding for each letter.

M	N	O	U	I	E	A
14	20	30	71	72	75	180

20. Insert the following integer keys into min-heap. Show the min-heap after each insertion.

65, 49, 81, 27, 88, 45, 36

21. Write a C function to compute the total number of external nodes in a binary search tree.

```
struct node {
      int data;
      struct node* leftChild;
      struct node* rightChild;
};

int totalNumberOfExternalNodes(struct node* tree) {
      ...
}
```

22. Write a C function to print all the data values of nodes in a binary search tree using pre-order walk.

```
void printPreOrder(struct node* tree) {
      ...
}
```

23. Grandparent node (G) of a node (N) refers to the parent of N's parent (P), whereas uncle node (U) of a node (N) refers to the sibling of N's parent (P). Write the definition of the following C functions using the given node structure.

```
struct node {
      int data;
      struct node* leftChild;
      struct node* rightChild;
};

struct node* grand_parent(struct node* n) {
      ...
}

struct node* uncle(struct node* n) {
      ...
}
```

---

24. Insert the following integer keys into the hash table using two different methods. (m = 10)

35, 28, 19, 38, 39, 15, 76

(a) Quadratic Probing

$h(k, i) = [h'(k) + c_1 i + c_2 i^2] \bmod m$

where $c_1 = 2$, $c_2 = 2$, $h'(k) = k \bmod m$

0	
1	
2	
3	
4	
5	
6	
7	
8	
9	

(b) Double Hashing

$h(k, i) = [h_1(k) + i h_2(k)] \bmod m$

where $h_1 = k \bmod 10$, $h_2 = k \bmod 7$

0	
1	
2	
3	
4	
5	
6	
7	
8	
9	