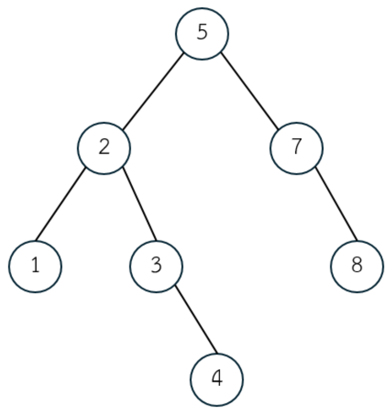
Conceptual Questions

[Some questions are modified from CP Data Structure Final Examination in previous years]

1. For the following binary tree, write down the order of nodes that traversal algorithms would visit them. Also, check if this binary tree is binary search tree.



Preorder Traversal	5 2 1 3 4 7 8
Inorder Traversal	1 2 3 4 5 7 8
Postorder Traversal	1 4 3 2 8 7 5
Is this a binary search tree?	Yes

2. Write down TRUE if the statement is true or FALSE is the statement is false in front of questions.

TRUE 2.1. Every AVL tree is binary search tree.

FALSE 2.2. Time complexity of searching a key in binary search tree is O(log n) where n is number of nodes in the tree. should be O(n)

TRUE 2.3. Binary heap can store duplicate values.

TRUE 2.4. Time complexity of searching value in AVL tree is O(log n) where n is number of nodes in the tree.

FALSE 2.5. In hash table, the load factor is always less than or equal to 1. For separate chaining hash table, load factor can be more than 1

FALSE 2.6. Time complexity of adding data into binary heap is Θ(log n). should be O(log n)

we usually ignore time complexity for expanding array when array is full because its average case is O(1)
[If we don't ignore it, it will be O(n)]

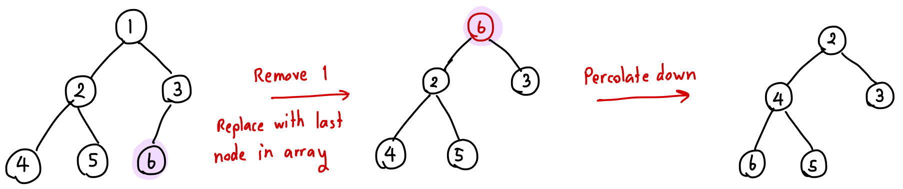
TRUE 2.7. Time complexity of merge sort is O(n log n).

FALSE 2.8. Time complexity of bubble sort is O(n log n). $O(n^2)$

FALSE 2.9. Time complexity of selection sort is O(n log n). $O(n^2)$

TRUE 2.10. There is no sorting algorithm that has time complexity of O(n).

3. If we insert 1, 2, 3, 4, 5, and 6 to empty binary min heap, then pop the value out for 1 time, draw the final binary min heap.



4. Consider an open addressing hash table which uses quadratic probing to solve collision problems. If the length of the array inside the has table is 13 and the hash function used in this table is

$$h(x) = x \% 13$$

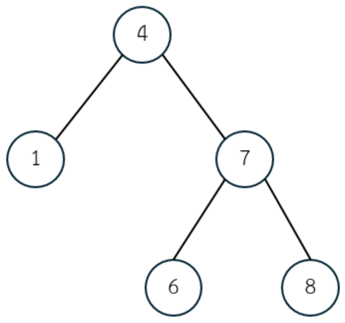
Write down the data in the hash table after performing the following operations. (write X if the data in that index is deleted)

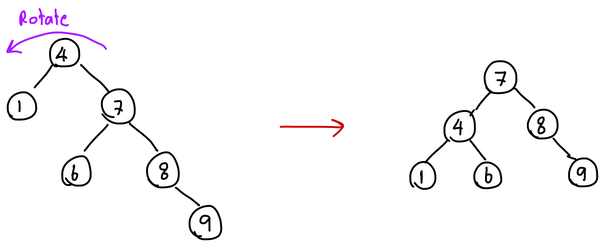
Operations	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]
(Example) Add 10											10		
(Example) Add 24											10	24	
(Example) Delete 10											X	24	
Add 1		1									X	24	
Add 11		1									X	24	11
Delete 24		1									X	X	11
Add 37		1									X	37	11
Add 23		1									23	37	11

5. Why must the size of array in hash tables be prime number?

Because the factors of prime number are 1 and itself, this will reduce a chance for colliding. If we use composite number, it may lead to more collisions if the keys being hashed are also multiple of the factors of that number.

6. From the given AVL tree, draw the new AVL tree after inserting value 9 into this tree.



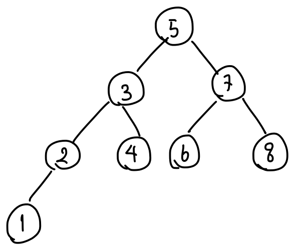


7. What is the purpose of AVL tree? Why don't we use normal binary search tree instead?

The purpose of AVL tree is to maintain a BST to ensure that insertion, searching, and deletion are performed with time complexity of O(log n), even in the worst case.

For BST, searching, insertion, and deletion degrade to O(n) instead.

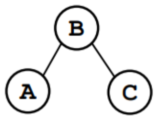
8. If we want to insert the value 1,2,3,4,5,6,7,8 into the empty AVL tree so that there is no any rotation occurred while inserting, write down the order of inserting (e.g. 5,6,3,4,2,1,8,7).



One possible answer is 5, 3, 7, 2, 4, 6, 8, 1

(There are many possible answers for this question)

9. Draw a binary tree that has the order of visiting of inorder and postorder traversals as shown below.

(Example) Inorder : A, B, C Postorder : A, C, B	
Inorder: X, Z, Y, U, V, W Postorder: V, U, W, Y, Z, X