Quiz 2

Section 1 (LEC0101, LEC2003)

Question 1

Consider a hash table with 100 slots. Collisions are resolved using chaining. Assuming simple uniform hashing, what is the probability that the first 2 slots are unfilled after the first 2 insertions?

A) $(99 \times 98)/100^2$

B) $99 \times 99/100^2$

C) $1/100$

D) $(98 \times 98)/100^2$

Question 1

Consider a hash table with 100 slots. Collisions are resolved using chaining. Assuming simple uniform hashing, what is the probability that the first 2 slots are unfilled after the first 3 insertions?

A) $(99 \times 98)/100^2$

B) $99 \times 99/100^2$

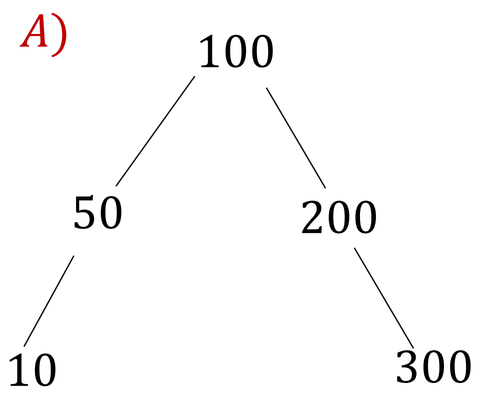
C) $1/100$

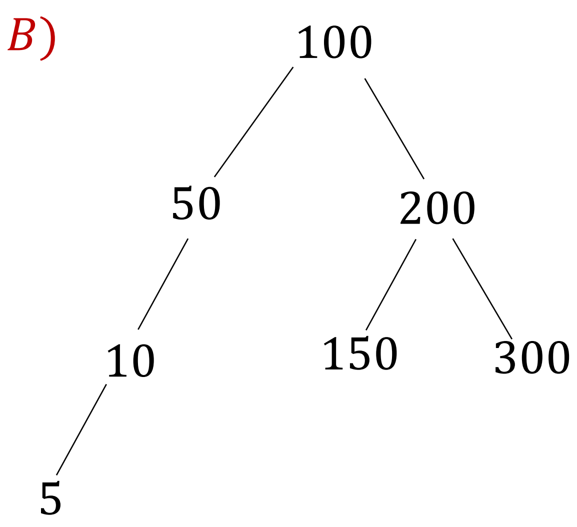
D) $(98 \times 98)/100^2$

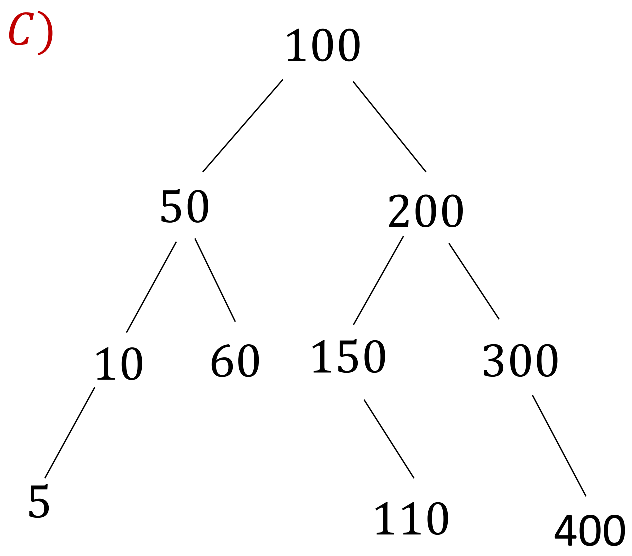
Probability that the first 2 slots are unfilled after the first 2 insertions = (probability that first item doesn't go in any of the first 2 slots)* (probability that second item doesn't go in any of the first 2 slots $(98/100) * (98/100)$

Question 2

Which of the following is AVL Tree?







A) Only A

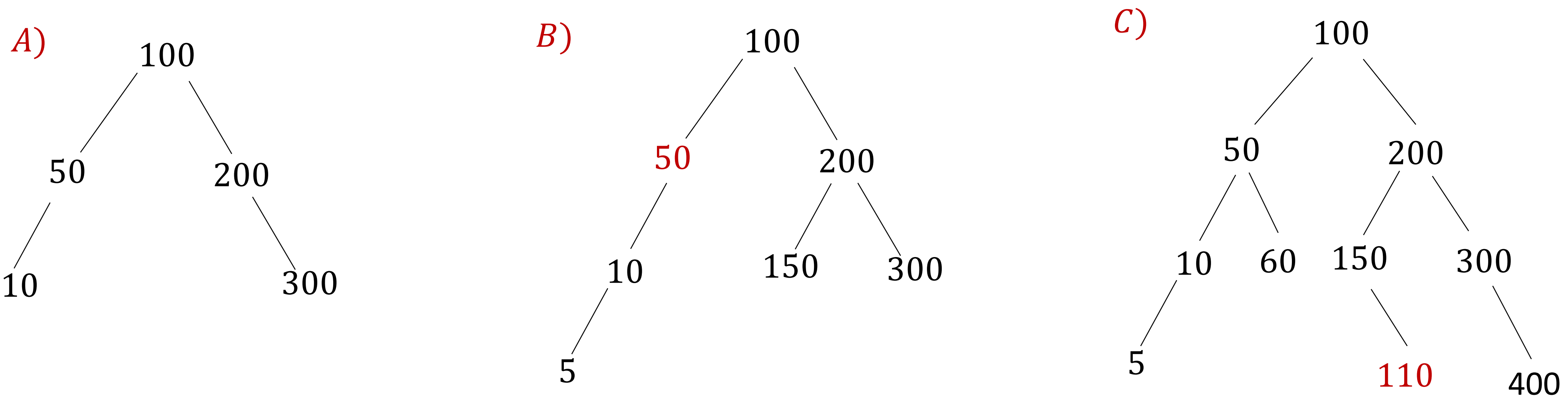
B) A and C

C) A, B and C

D) Only B

Question 2

Which of the following is AVL Tree?



A) Only A

B) A and C

C) A, B and C

D) Only B

A Binary Search Tree is AVL if balance factor of every node is either -1 or 0 or 1. Balance factor of a node X is [(height of X->left) - (height of X->right)]. In Tree B, the node with value 50 has balance factor 2. That is why B is not an AVL tree. C is not a BST, 150>100

Question 3

To balance an AVL tree aftre an insertion, how many number of rotations is requiered? What is the time complexity of a rotation?

A) Two rotations, $O(\log n)$

B) One rotation, $O(\log n)$

C) $O(\log n)$ rotations, $O(1)$

D) Two rotations, $O(1)$

Question 3

To balance an AVL tree aftre an insertion, how many number of rotations is requiered? What is the time complexity of a rotation?

A) Two rotations, $O(\log n)$

B) One rotation, $O(\log n)$

C) $O(\log n)$ rotations, $O(1)$

D) Two rotations, $O(1)$

After any insertion, a single or a double hashing would balance the tree, any rotation modifies constants and only two nodes' height would change, so the time complexity of a rotation is O(1).

Question 4

Suppose that each cell in a hash table stores a Binary search tree instead of a linked list. What would be the worst case running time of search? (n is the number of elements)

A) $O(1)$

B) $O(n)$

C) $O(\log n)$

D) $O(\sqrt{n})$

Question 4

Suppose that each cell in a hash table of size m stores a Binary search tree instead of a linked list. What would be the worst case running time of search?

A) $O(1)$

B) $O(n)$

The height of a BST is O(n) in the worst case.

C) $O(\log n)$

D) $O(\sqrt{n})$

Question 5

For which data structure, the order of insertion does not matter, i.e. the resulting data structure is identical regardless of the order the elements were inserted?

A) Heap

B) AVL tree

C) Hash table with chaining

D) Sorted array

Question 5

For which data structure the order of insertion does not matter, i.e. the resulting data structure is identical regardless of the oder the elements were inserted?

A) Heap

B) AVL tree

C) Hash table with chaining

D) Sorted array

All other structures have more than possible structure with the same elements.

Question 6

What is the number of possible binary search trees that can be created with 3 items $a < b < c$?

A) There is only a unique BST

B) 3

C) 5

D) $2^3 = 8$

Question 6

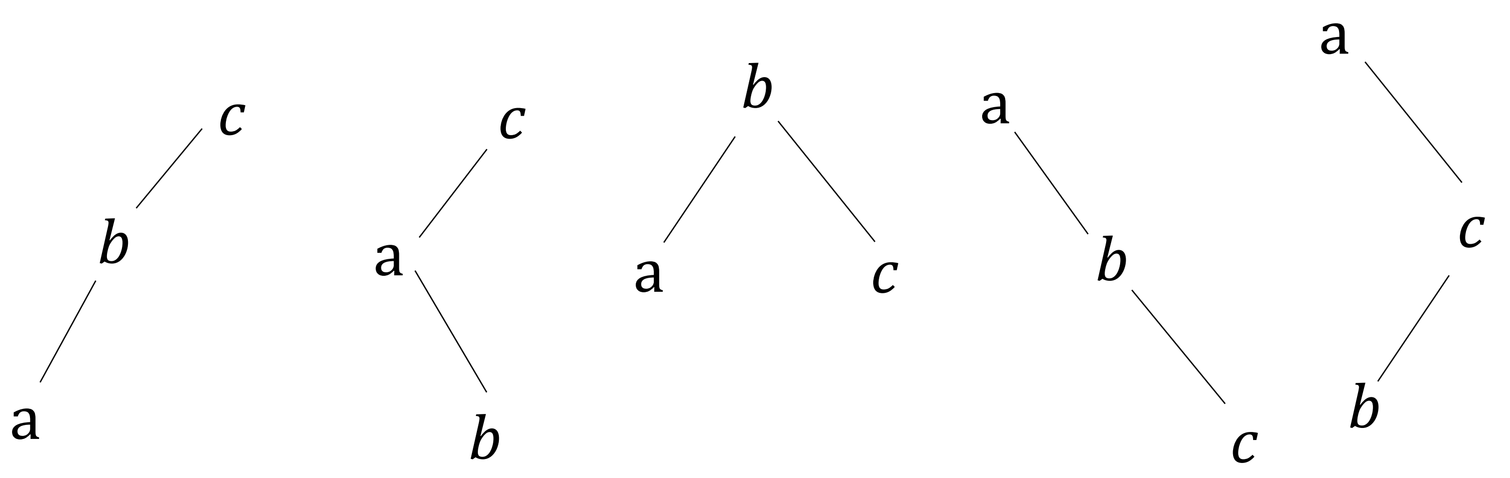
What is the number of possible binary search trees that can be created with 3 items $a < b < c$?

A) There is only a unique BST

B) 3

C) 5

D) $2^3 = 8$



Question 7

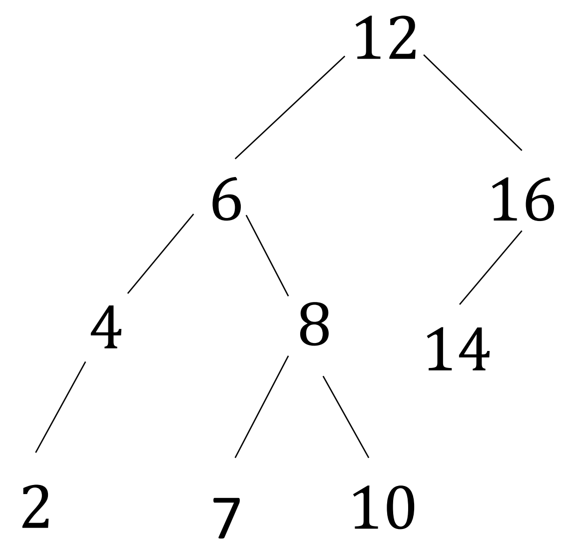
We want to insert 9 into the AVL tree shown below. What is the pivot node (or nodes) about which rotation should be performed to restore the AVL tree property after insertion?

A) 8

B) 12

C) 6, 12

D) 6, 8



Question 7

We want to insert 9 into the AVL tree shown below. What is the pivot node (or nodes) about which rotation should be performed to restore the AVL tree property after insertion?

A) 8

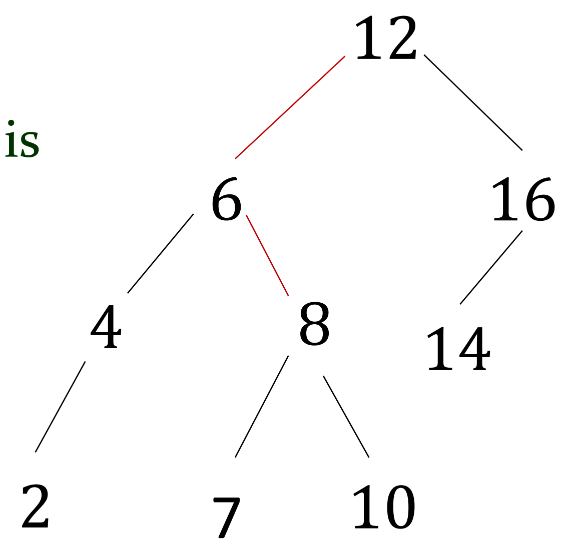
B) 12

C) 6, 12

D) 6, 8

12 is the lowest node which is imbalanced.

It is left heavy, needs a right rotation but since 6 is also right heavy, it needs first to perform a left rotation on 6.



Question 8

Consider a hash table with chaining that contains 1000 elements. Let x_{100} is the 100^{th} element that is inserted in the table. What is the maximum number of elements need to be looked up for search x_{100}?

A) 1000

B) 101

C) 10

D) 901

Question 8

Consider a hash table with chaining that contains 1000 elements. Let x_{100} is the 100^{th} element that is inserted in the table. What is the maximum number of elements need to be looked up for search x_{100}?

A) 1000

B) 101

C) 10

D) 901

Since the elements are inserted at the head of the chain, so it is not possible that anything that is inserted before x_{100} to be looked up. So the number of elements to find x_{100} is maximum 901.

$$x_{1000} \rightarrow x_{999} \rightarrow \ \ldots\ldots -> x_{100} \rightarrow \ \ldots . x_1$$

Question 9

What is the expected number of comparisons for unsuccessful search in a hash table of size 15 in which 30 keys are stored, assuming that the chaining collision resolution mechanism is used?

A) 30

B) 15

C) 2

D) 0.5

Question 9

What is the expected number of comparisons for unsuccessful search in a hash table of size 15 in which 30 keys are stored, assuming that the chaining collision resolution mechanism is used?

A) 30

B) 15

C) 2

D) 0.5

Load factor: $\alpha = \frac{n}{m} = \frac{30}{15} = 2$

Question 10

For a hash function $h(k) = k \ mod \ m$, which one is a best choice for m?

$A) \ m = 32$

$B) \ m = 17$

$C) \ m = 29$

$D) \ m = 33$

Question 10

For a hash function $h(k) = \; k \; mod \; m$, which one is a better choice for m?

A) $m = 32$

B) $m = 17$

C) $m = 29$

D) $m = 33$

29 is a prim number and not close to power of 2