Who wants to be a ~~millionaire~~ Computer Scientist?

Preparation for the finals

Game 2

Question 1. 500 points

- Identify the code which **does not** run in time O(1)

A.

```
for (i = 0; i < 10; i++)
   sum += num2;
```

B.

```
if (x > y)
   return x;
```

C.

```
i = 0;
while (i < listSize) {
   sum = sum + l;
   i++;
}
```

D.

```
num = arr[i];
arr[i + 1] = num + 1;
```

A	
B	

C	D
D	B

Question 1. 500 points

- Identify the code which **does not** run in time O(1)

A.

```
for (i = 0; i < 10; i++)
   sum += num2;
```

B.

```
if (x > y)
   return x;
```

C.

```
i = 0;
while (i < listSize) {
   sum = sum + l;
   i++;
}
```

D.

```
num = arr[i];
arr[i + 1] = num + 1;
```

A
B

C
D

The correct answer is C.

Question 2. 1,000 points

- Which of the following is an example of constant time O(1) ?

A. Finding the minimum value of an array
B. Binary search
C. Accessing an element of an array
D. Bubble sort

A	
B	

C	
D	

Question 2. 1,000 points

- Which of the following is an example of constant time O(1) ?

A. Finding the minimum value of an array
B. Binary search
C. Accessing an element of an array
D. Bubble sort

A	C
B	D

The correct answer is C.

Question 3. 2,000 points

- What is the complexity of *heapsort*?

A	O (n)
B	O (n log n)

C	O (log n)
D	O (n^2)

Question 3. 2,000 points

- What is the complexity of *heapsort*?

A	O (n)
B	O (n log n)

C	O (log n)
D	O (n^2)

The correct answer is B.

Question 4. 3,000 points

- *ListTraverseReverse* must print a linked list in reverse order. Which XXX should replace the missing statement?

```
ListTraverseReverse(list) {
   XXX
}


printReverse(node) {
   if (node is not null) {
      printReverse(node⇢next)
      print node
   }
}
```

A. `printReverse(list⇢head)`
B. `printReverse(list)`
C. `printReverse(list⇢head⇢next)`
D. `printReverse(list⇢tail)`

A	
B	

C	
D	

Question 4. 3,000 points

- *ListTraverseReverse* must print a linked list in reverse order. Which XXX should replace the missing statement?

```
ListTraverseReverse(list) {
   XXX
}


printReverse(node) {
   if (node is not null) {
      printReverse(node⇢next)
      print node
   }
}
```

A. `printReverse(list⇢head)`
B. `printReverse(list)`
C. `printReverse(list⇢head⇢next)`
D. `printReverse(list⇢tail)`

A	C
B	D

The correct answer is A.

Question 5. 5000 points

- The algorithm performs exactly $7+12N+3N^2$ steps. What is the Big O of this algorithm?

A	$3N^2$
B	N^3

C	N^2
D	$12N$

Question 5. 5000 points

- The algorithm performs exactly $7+12N+3N^2$ steps. What is the Big O of this algorithm?

A	$3N^2$	C	N^2
B	N^3	D	$12N$

The correct answer is C.

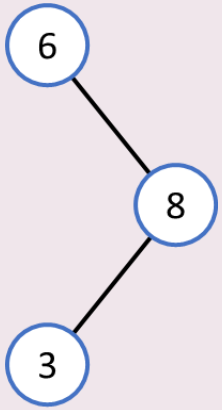
Checkpoint 1 reached!

You have 5,000 points

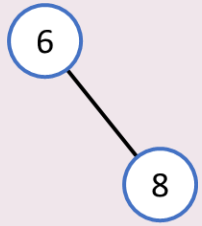
Question 6. 7,500 points

- Which of the following is an AVL tree?

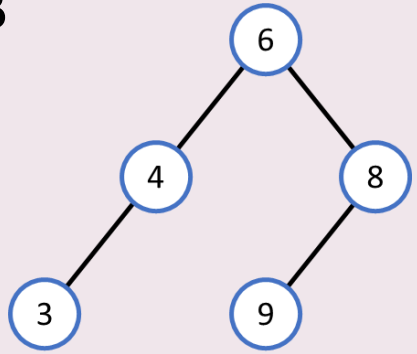
A



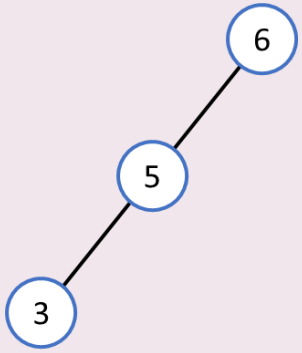
C



B



D



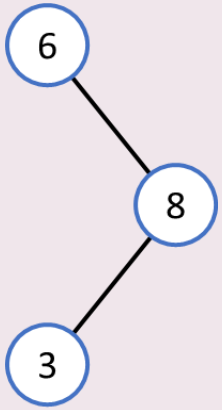
A	
B	

C	
D	

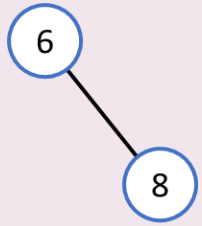
Question 6. 7,500 points

- Which of the following is an AVL tree?

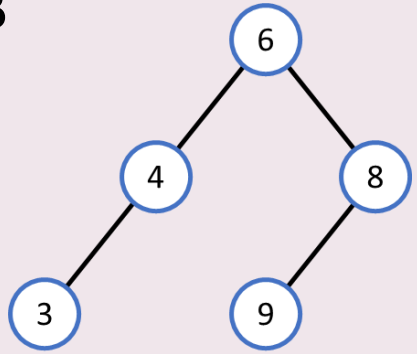
A



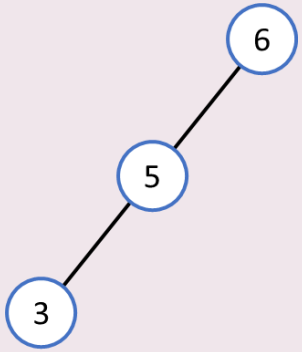
C



B



D



A	C
B	D

The correct answer is C.

Question 7. 10,000 points

- Which of the following is TRUE?

A. The cost of searching a binary search tree is O(log n) and that of an AVL tree is O(n)
B. The cost of searching a binary search tree is O(n) and that of an AVL tree is O(log n)
C. The cost of searching a binary search tree is O(log n) and that of an AVL tree is O(log n)
D. The cost of searching a binary search tree is O(n) and that of an AVL tree is O(n)

A	
B	

C	
D	

Question 7. 10,000 points

- Which of the following is TRUE?

A. The cost of searching a binary search tree is O(log n) and that of an AVL tree is O(n)
B. The cost of searching a binary search tree is O(n) and that of an AVL tree is O(log n)
C. The cost of searching a binary search tree is O(log n) and that of an AVL tree is O(log n)
D. The cost of searching a binary search tree is O(n) and that of an AVL tree is O(n)

A	C
B	D

The correct answer is B.

Question 8. 15,000 points

- Which is an Abstract data type (ADT)?

A	Linked List
B	Stack

C	Adjacency matrix
D	None of the above

Question 8. 15,000 points

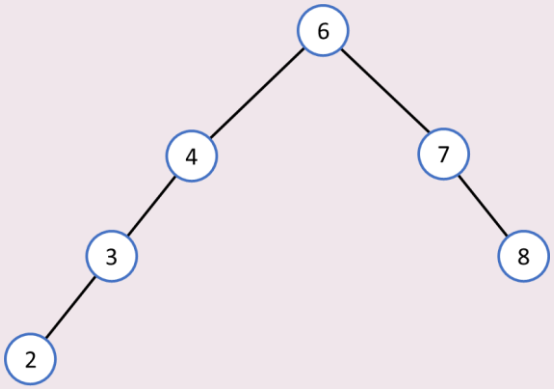
- Which is an Abstract data type (ADT)?

A	Linked List	C	Adjacency matrix
B	Stack	D	None of the above

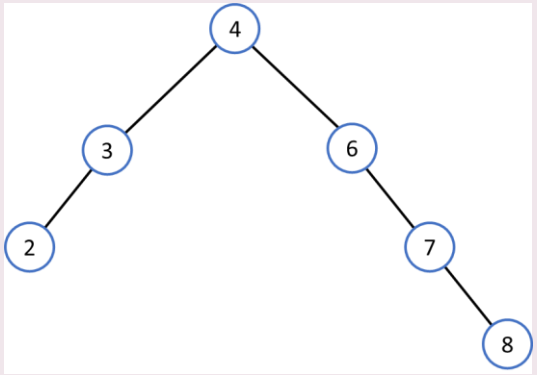
The correct answer is B.

Question 9. 25,000 points

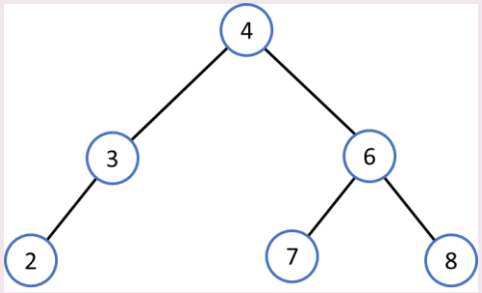
- What is the result of rebalancing the following AVL tree?



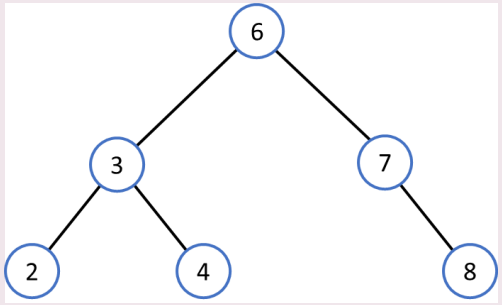
A



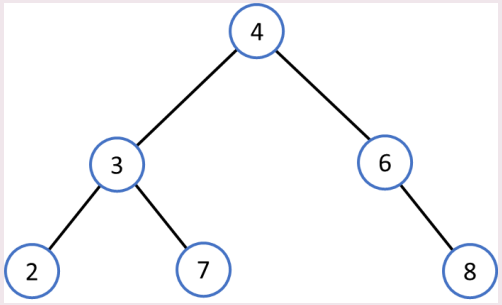
C



B



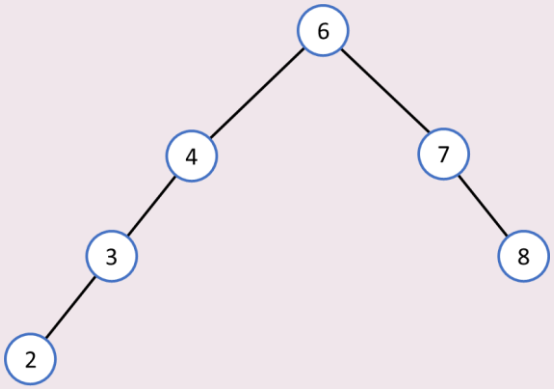
D



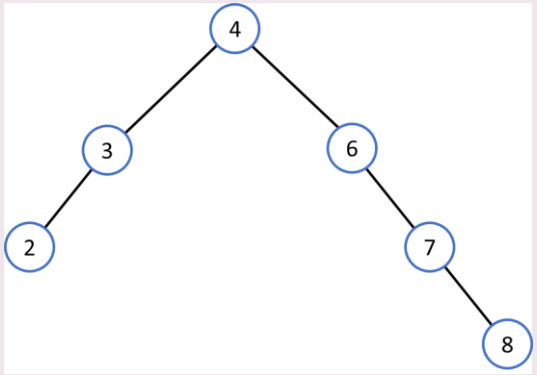
A		C	
B		D	

Question 9. 25,000 points

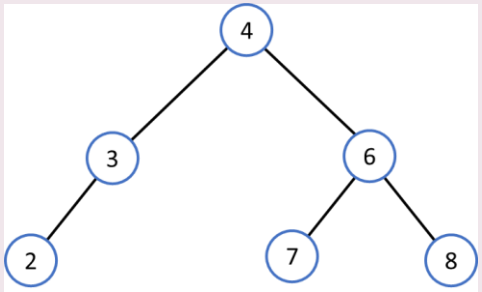
- What is the result of rebalancing the following AVL tree?



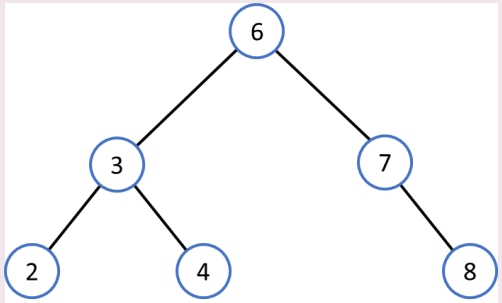
A



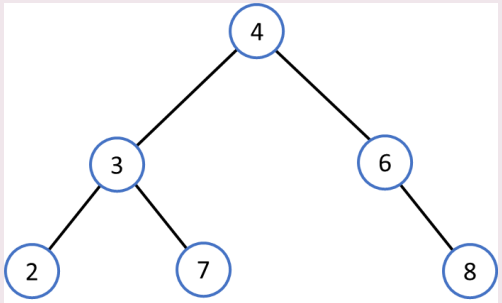
C



B



D



A	C
B	D

The correct answer is B.

Question 10. 50,000 points

- If the *binarySearch()* method is called to search a sorted array of 32 numbers, then at most _____ array numbers are compared against the search key.

A	6
B	32

C	4
D	5

Question 10. 50,000 points

- If the *binarySearch()* method is called to search a sorted array of 32 numbers, then at most _____ array numbers are compared against the search key.

A 6	C 4
B 32	D 5

The correct answer is D.

Checkpoint 2 reached!

You have 50,000 points

Question 11. 75,000 points

- Which XXX completes the *append*() method in the Java LinkedList class for a singly-linked list?

```
public void append(Node newNode) {
   if (head == null) {
      head = newNode;
      tail = newNode;
   }
   else {
      XXX
      tail = newNode;
   }
}
```

```
A.  head.next = newNode;
B.  head = newNode;
C.  tail.next = newNode;
D.  head = tail;
```

A	
B	

C	
D	

Question 11. 75,000 points

- Which XXX completes the *append*() method in the Java LinkedList class for a singly-linked list?

```
public void append(Node newNode) {
   if (head == null) {
      head = newNode;
      tail = newNode;
   }
   else {
      XXX
      tail = newNode;
   }
}
```

```
A.  head.next = newNode;
B.  head = newNode;
C.  tail.next = newNode;
D.  head = tail;
```

A	C
B	D

The correct answer is C.

Question 12. 150,000 points

- What is the height of a BST built by inserting nodes in the order 12, 24, 23, 48, 47?

A	4
B	3

C	1
D	2

Question 12. 150,000 points

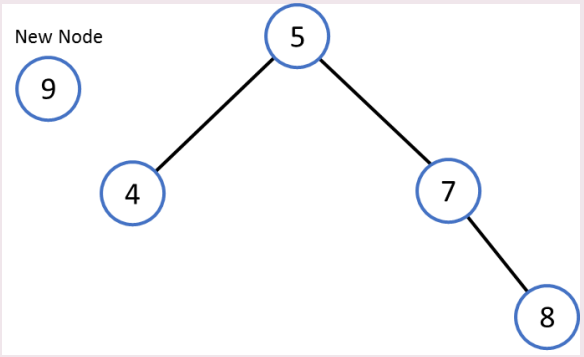
- What is the height of a BST built by inserting nodes in the order 12, 24, 23, 48, 47?

A	4	C	1
B	3	D	2

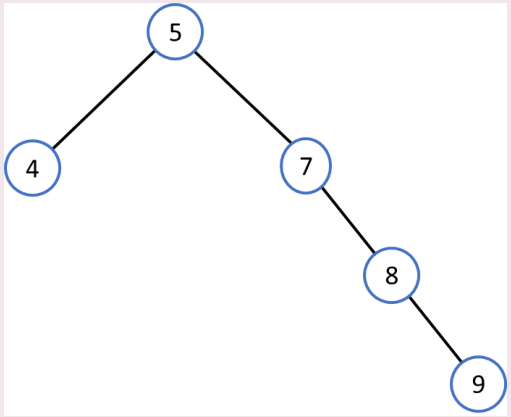
The correct answer is B.

Question 13. 250,000 points

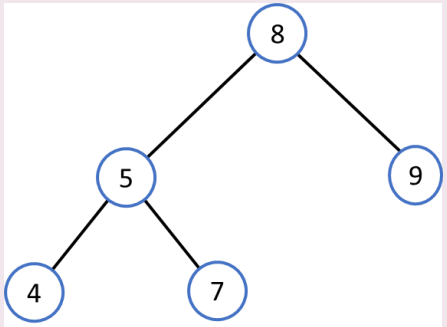
- Identify the AVL tree which results after insertion of node 9 into the following tree:



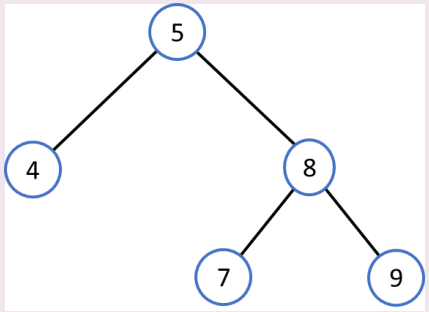
A



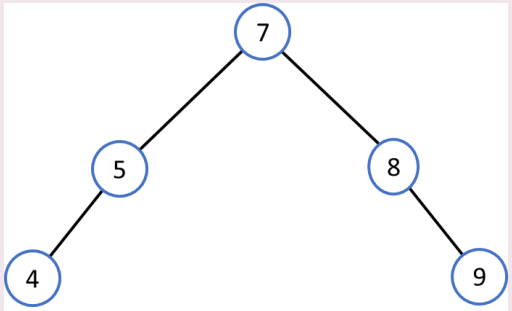
C



B



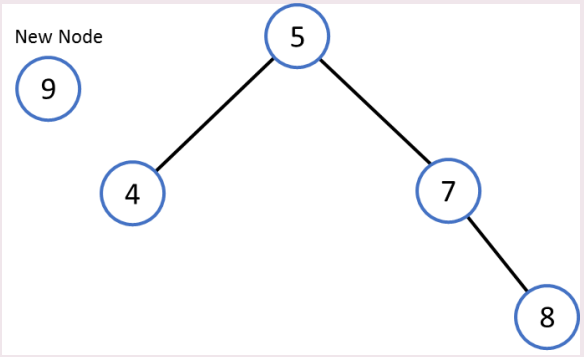
D



A	
B	

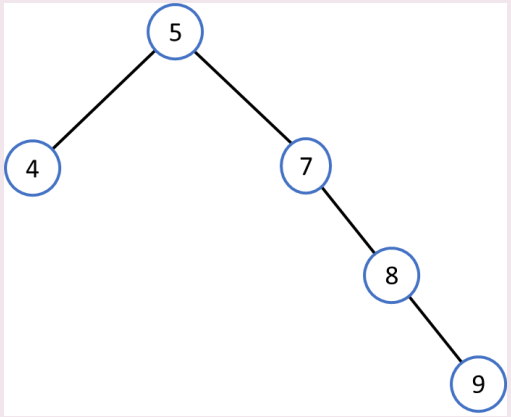
C	
D	

Question 13. 250,000 points

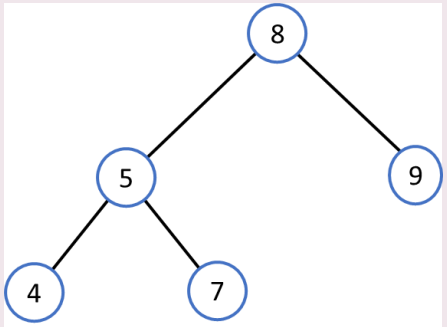


- Identify the AVL tree which results after insertion of node 9 into the following tree:

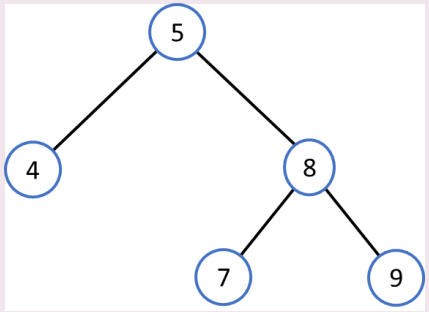
A



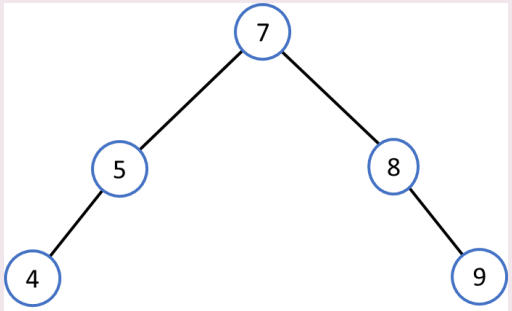
C



B



D



A	C
B	D

The correct answer is B.

Question 14. 500,000 points

- The queue was implemented using a circular array.
- What is the condition XXX?

```
enqueue(element, Array A, read, write)
   if XXX
      print ("Queue is full")
      return
   A[write] = element
   write = (write+1) % A.length
```

A. write == read
B. write == (read + 1) % A.length
C. write == (read – 1) % A.length
D. write == read - 1

A	
B	

C	
D	

Question 14. 500,000 points

- The queue was implemented using a circular array.
- What is the condition XXX?

```
enqueue(element, Array A, read, write)
   if XXX
      print ("Queue is full")
      return
   A[write] = element
   write = (write+1) % A.length
```

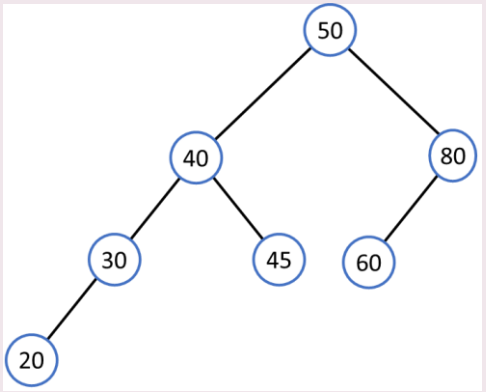
A. write == read
B. write == (read + 1) % A.length
C. write == (read – 1) % A.length
D. write == read - 1

A	C
B	D

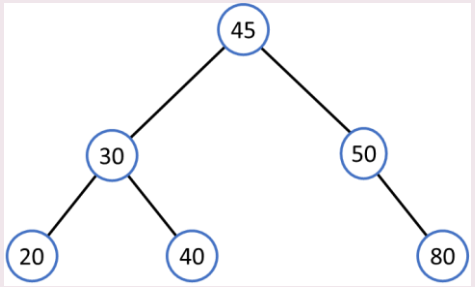
The correct answer is C.

Question 15. **One million points!**

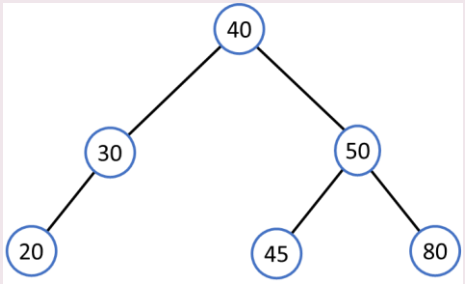
- Identify the rebalanced AVL tree after removing 60 from the following tree:



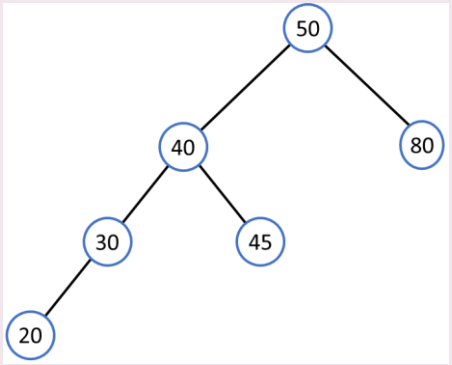
A



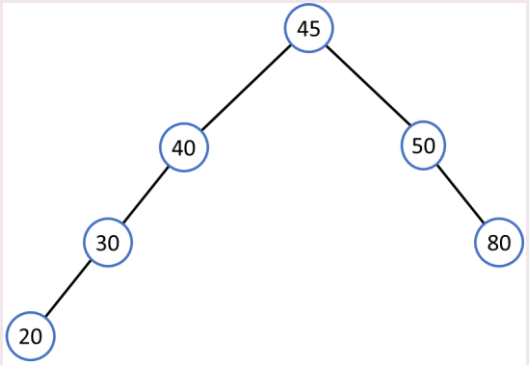
C



B



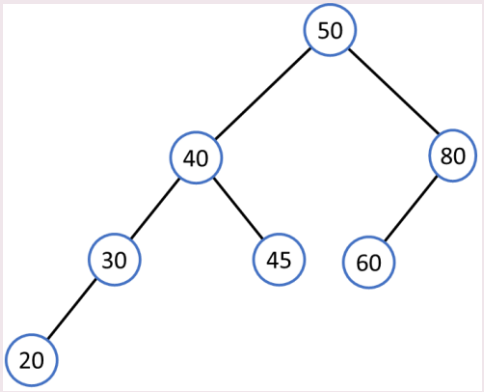
D



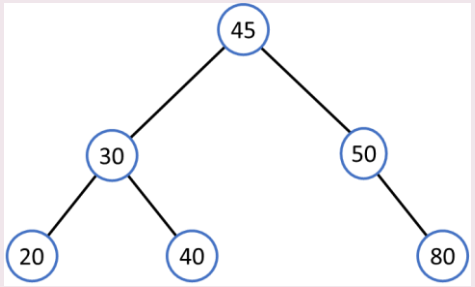
A		C	
B		D	

Question 15. **One million points!**

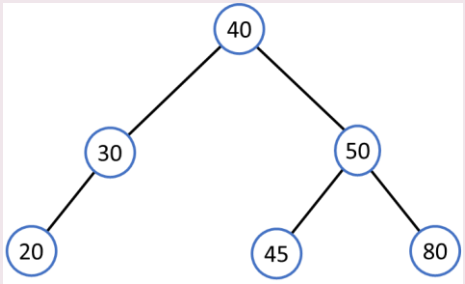
- Identify the rebalanced AVL tree after removing 60 from the following tree:



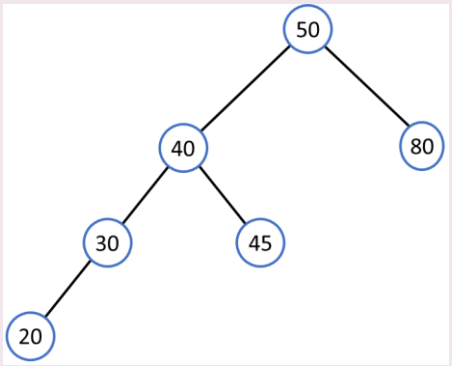
A



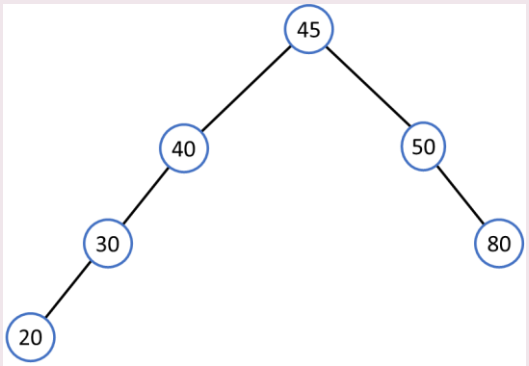
C



B



D



A	C
B	D

The correct answer is C.

Well done!

You are almost ready for the final exam