M13.D2: Exam: Data Structures Final Exam (in-Class Exam)

Due May 10, 2022 at 2pm **Points** 30 **Questions** 30
Available after May 10, 2022 at 1pm **Time Limit** 60 Minutes

This quiz is no longer available as the course has been concluded.

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	17 minutes	30 out of 30

Correct answers are hidden.

Score for this quiz: **30** out of 30
Submitted May 10, 2022 at 1:18pm
This attempt took 17 minutes.

Question 1 — 1 / 1 pts

Which of the following activities can be accomplished using recursive algorithms?

- Making lemonade
- Harvesting crops

- Writing a letter to a friend
- Going to the supermarket to buy groceries

Question 2

1 / 1 pts

Which XXX will complete the code?

```
public class RecursiveCalls {
   public static int convertToBinary(int num) {
       if (num == 0) {
          return 0;
       }
       else {
          return (num % 2 + 10 * XXX);
       }
   }
   public static void main (String [] args) {
      System.out.print("The Binary number is: ");
      System.out.println(convertToBinary(7));
   }
}
```

- convertToBinary(num * 2);
- convertToBinary(num / 2);
- convertToBinary();
- convertToBinary(num);

Question 3

1 / 1 pts

Identify the base case in the following code.

```
public class FindMatch {
   public static int findMatch(char array[], int low, int high, cha
r key) {
      if (high >= low) {
         int mid = low + (high - low) / 2;
         if (array[mid] == key) {
            return mid;
         }
         if (array[mid] > key) {
            return findMatch(array, low, mid, key);
         }
         else {
            return findMatch(array, mid + 1, high, key);
         }
      }
      return -1;
   }
}
```

- int mid = low + (high - low)/2;
- if (high>=low)
- if (array[mid] > key)
- if (array[mid] == key)

Question 4

1 / 1 pts

In the following code, the variable indentAmt _____ at each recursive call.

```
public static int sum(int n, String indentAmt) {
      int sum = 0;
      if (n == 0) {
         System.out.println(indentAmt + "your value is: " + n);
         return 0;
      }
      else {
         System.out.println(indentAmt + "your value is: " + n);
         sum = n + sum(n - 1, indentAmt + "   ");
      }
      System.out.println(indentAmt + "Returning pos = " + n);
      return sum;
   }
```

- is decremented
- remains constant
- is incremented
- is incremented or decremented based on the call

Question 5

1 / 1 pts

What is output?

```
public class RecursionExample {
   public static void printer() {
      System.out.println("hello");
      printer();
   }


   public static void main(String[] args)  {
      printer();
   }
}
```

- hello
- hello
  hello
- stack overflow
- hello
  hello
  hello

Question 6

1 / 1 pts

A list of employees that has been sorted in descending order needs to be reversed. Which XXX completes the algorithm to sort the list in ascending order?

```
AscendingList(empList, begin, end) {
   if (begin >= end)
      return
   else {
      Swap empList[begin] and empList[end]
      XXX
   }
}
```

- (•) AscendingList(empList, begin + 1, end - 1)
- ( ) AscendingList(empList, begin - 1, end + 1)
- ( ) AscendingList(empList, end, begin)
- ( ) AscendingList(empList, begin, end)

Question 7

1 / 1 pts

Which explanation matches the following runtime complexity?
$T(N) = k + T(N-1)$

- (•) Every time the function is called, k operations are done, and the recursive call lowers N by 1.
- ( )

Every time the function is called, k operations are done, and the recursive call lowers N by k.

Every time the function is called, k operations are done, and each recursive call lowers N by one fourth.

Every time the function is called, k operations are done, and each of the 2 recursive calls reduces N by half.

Question 8

1 / 1 pts

What is the Big O notation for a recursive function with a runtime complexity of $T(N) = 5N + T(N-1)$?

- $O(N^2 \cdot logN)$
- $O(N^2)$
- $O(N \cdot logN)$
- $O(5N^2 \cdot logN)$

Question 9

1 / 1 pts

Which XXX should replace the missing statement in the following algorithm?

```
ListSearch(myData, key) {
   return ListSearchRecursive(key, myData→head)
}


ListSearchRecursive(key, node) {
   if (node is not null) {
      XXX {
         return node
      }
      return ListSearchRecursive(key, node→next)
   }
   return null
}
```

- if (node→tail == key)
- if (node→head == key)
- if (node→data == key)
- if (node→next == key)

Question 10

1 / 1 pts

Identify the correct statement about binary space partitioning.

- In animation only one region is analyzed at a time.
-

Half the objects are eliminated each level while traversing down a BSP tree.

Regions are always split down the middle either horizontal or vertical.

BSP tree can be used to store all objects in a two-dimensional world.

Question 11

1 / 1 pts

Which XXX completes the BST search algorithm?

```
BSTSearch(tree, key) {
  cur = tree→root
  while (cur is not null)
     if (key == cur→key)
        return cur
     else XXX
        cur = cur→left
     else
        cur = cur→right
  return null
}
```

- if (key < cur⋯→key)
- if (key < cur⋯→left⋯→key)
- if (key > cur⋯→key)
- if (key > cur⋯→right⋯→key)

Question 12

1 / 1 pts

Which XXX completes the BST in order traversal algorithm?

```
BSTPrintInorder(node) {
   if (node is null)
      return


   XXX
   Print node
   BSTPrintInorder(node→right)
}
```

- BSTPrintInorder(node→right)
- BSTPrintInorder(node)
- BSTPrintInorder(null)
- BSTPrintInorder(node→left)

Question 13

1 / 1 pts

Which XXX should replace the missing statement in the following BSTGetHeight algorithm?

```
BSTGetHeight(node){
   if (node is null)
      return -1
   leftHeight = BSTGetHeight(node→left)
   rightHeight = BSTGetHeight(node→right)
   XXX
}
```

- (•) return 1 + max(leftHeight, rightHeight)
- ( ) return max(leftHeight, rightHeight)
- ( ) return 1 + min(leftHeight, rightHeight)
- ( ) return min(leftHeight, rightHeight)

Question 14

1 / 1 pts

Which code block should be used to insert a new node in an empty tree?

- ( )
```
BSTInsert(tree, node) {
   if (node→right == null) {
      tree→root = node
      node→parent = node→left
      return
   }
}
```

```
BSTInsert(tree, node) {
   if (node→left == null) {
      tree→root = node
      node→parent = null
      return
   }
}
```

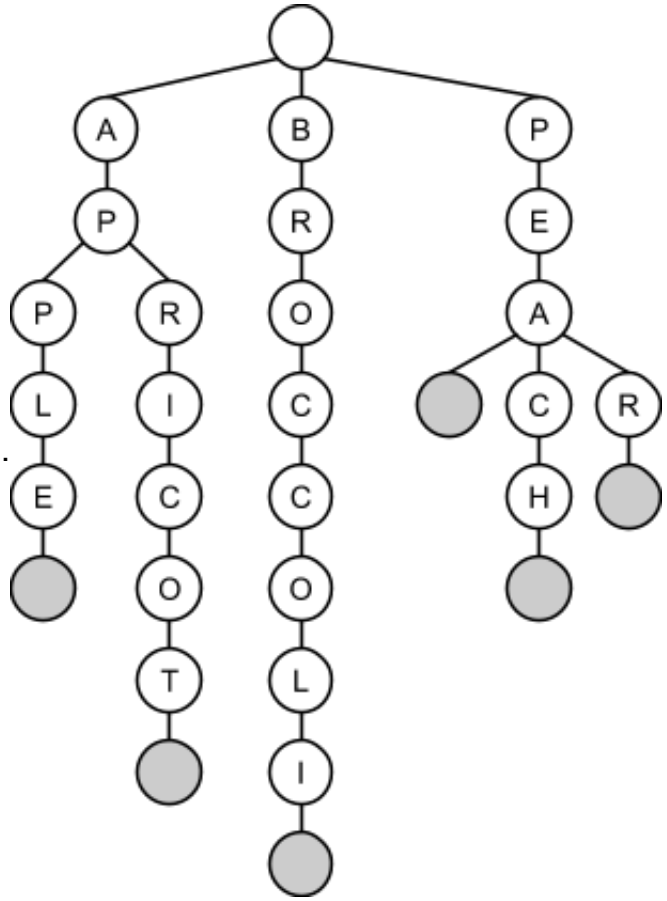
```
BSTInsert(tree, node) {
   if (tree→root == null) {
      tree→root = node
      node→parent = node→left
      return
   }
}
```

```
BSTInsert(tree, node) {
   if (tree→root == null) {
      tree→root = node
      node→parent = null
      return
   }
}
```

Question 15

1 / 1 pts

Calling TrieRemove(root, "PEACH") on the trie below removes _____ nodes from the trie.

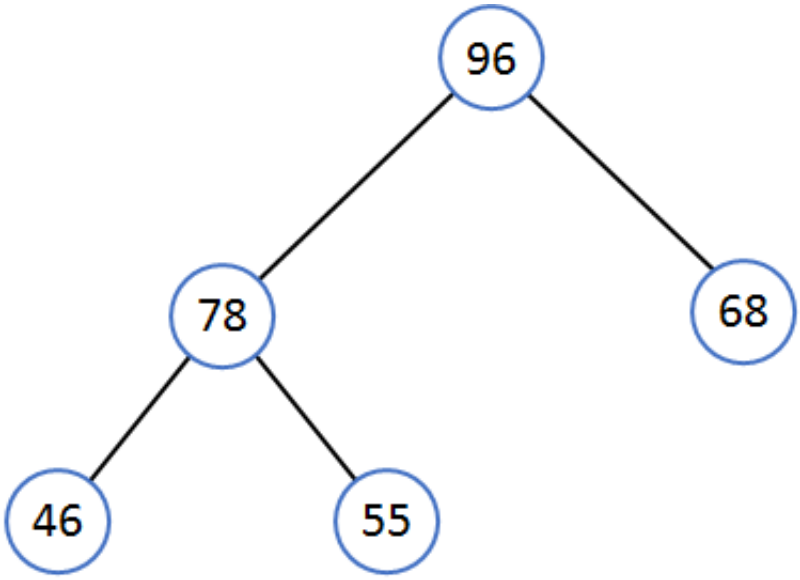


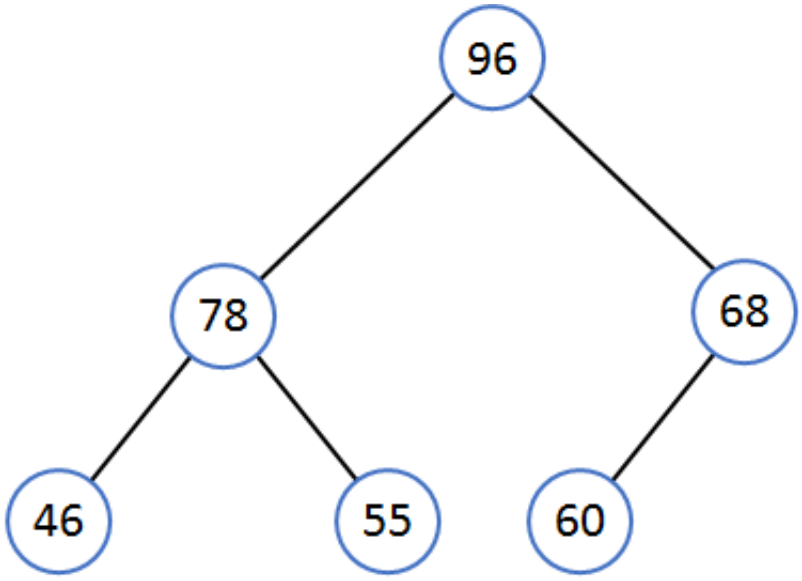
- 3
- 5
- 2
- 6

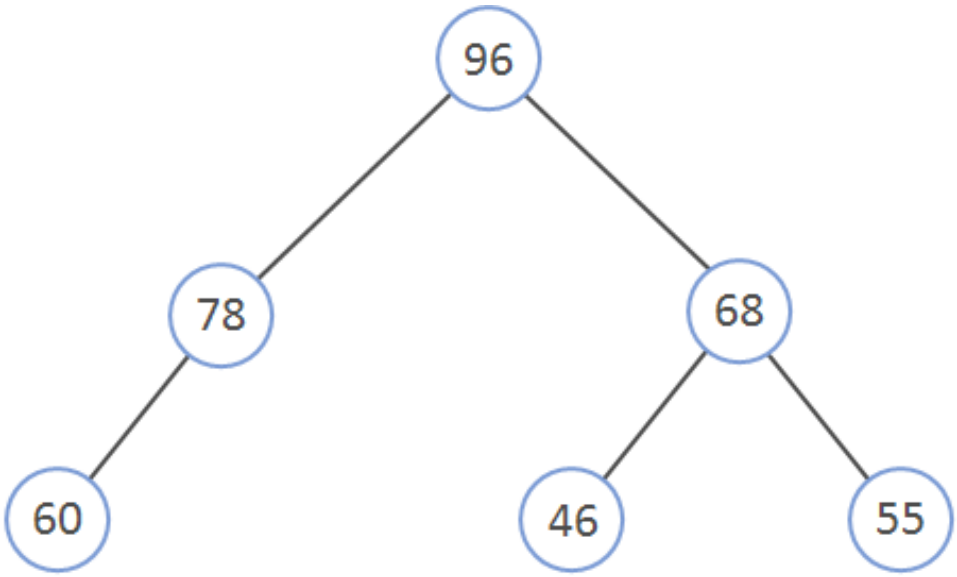
Question 16

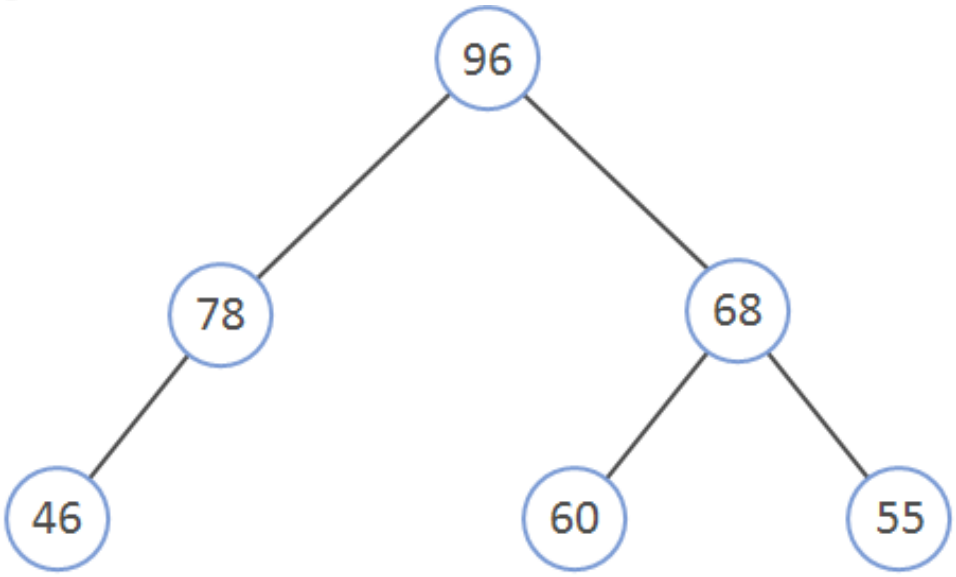
1 / 1 pts

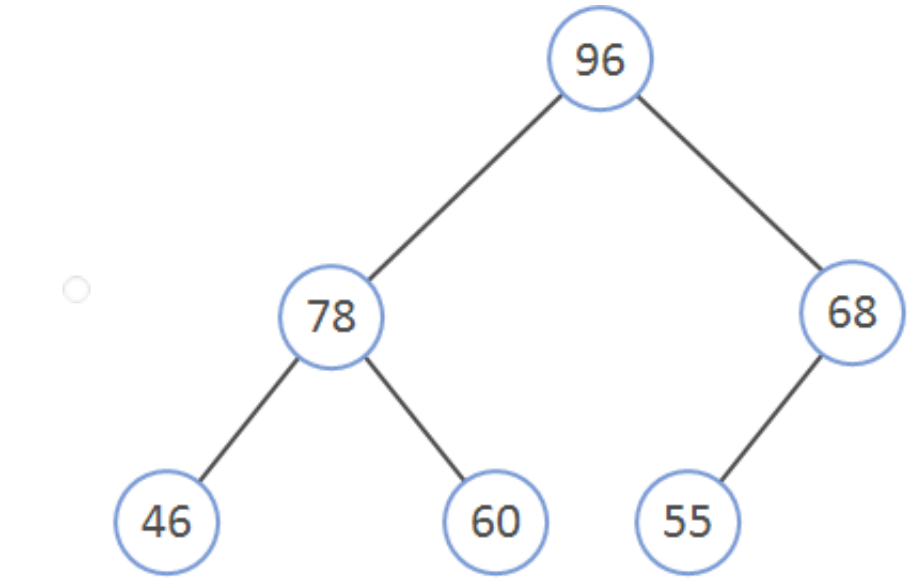
Identify the new binary max-heap after inserting 60 in the following max-heap.






96
78
68
60
46
55


96
78
68
46
60
55



Question 17

1 / 1 pts

Which XXX would replace the missing statement in the given MaxHeapPrecolateDown() function?

```
MaxHeapPercolateDown(nodeIndex, heapArray, arraySize) {
   childIndex = 2 * nodeIndex + 1
   value = heapArray[nodeIndex]
   while (childIndex < arraySize) {
      maxValue = value
      maxIndex = -1
      XXX {
         if (heapArray[i + childIndex] > maxValue) {
            maxValue = heapArray[i + childIndex]
            maxIndex = i + childIndex
         }
      }
      if (maxValue == value)
         return
      else {
         swap heapArray[nodeIndex] and heapArray[maxIndex]
         nodeIndex = maxIndex
         childIndex = 2 * nodeIndex + 1
      }
   }
}
```

- for (i = 0; i < 2 && i + childIndex < arraySize; i++)
- for (i = 0; i < 2 && i + childIndex > arraySize; i--)
- for (i = 0; i < 2 && i + childIndex > arraySize; i++)
- for (i = 3; i > 2 && i + childIndex < arraySize; i--)

Question 18

1 / 1 pts

Which XXX would replace the missing statement in the given Heapsort algorithm?

```
Heapsort(numbers, numbersSize) {
   XXX
      MaxHeapPercolateDown(i, numbers, numbersSize)

   for (i = numbersSize - 1; i > 0; i--) {
      swap numbers[0] and numbers[i]
      MaxHeapPercolateDown(0, numbers, i)
   }
}
```

- for (i = numbersSize / 2 - 1; i >= 0; i--)
- for (i = numbersSize+1; i >= 0; i--)
- for (i = numbersSize * 2; i >= 0; i--)
- for (i = numbersSize - 2; i >= 1; i--)

Question 19

1 / 1 pts

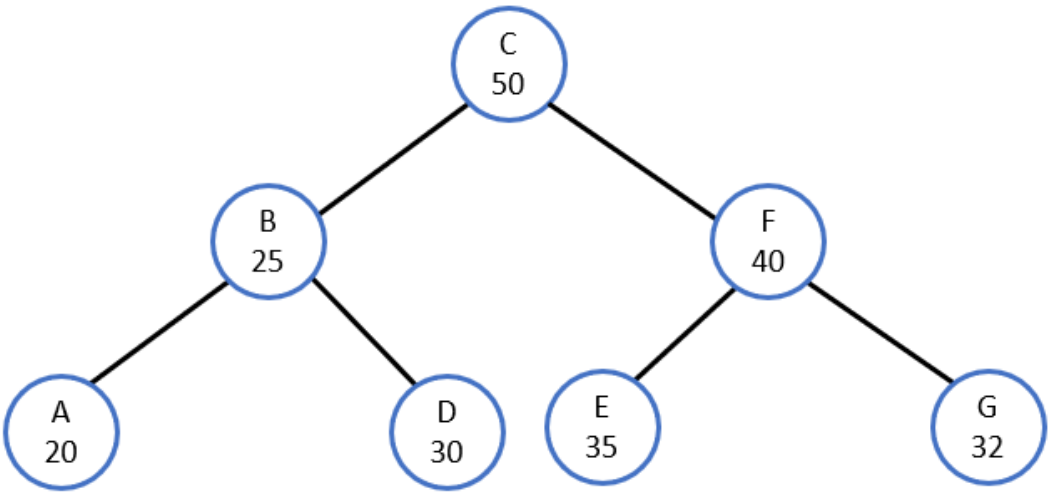
What is the worst-case runtime complexity of a peek operation on a priority queue?

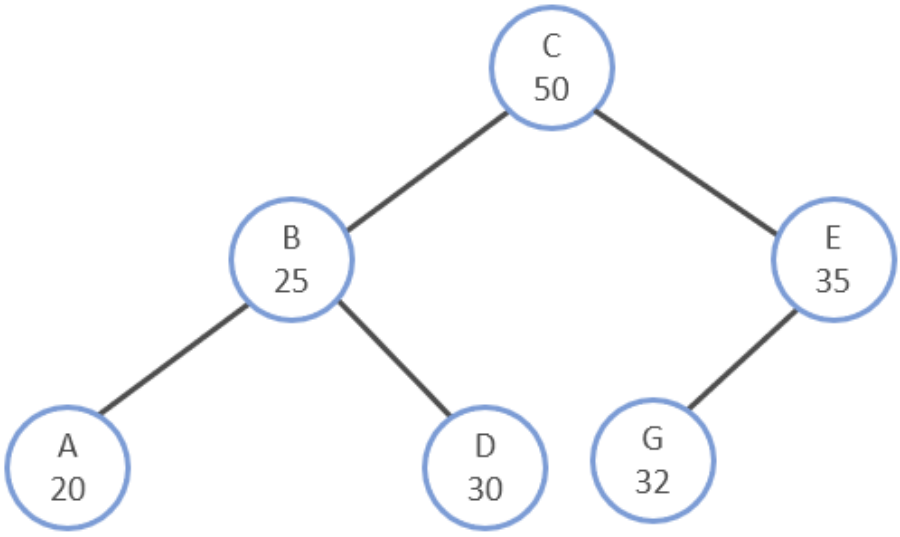
- $O(N)$
- $O(logN)$
- $O(NlogN)$
- $O(1)$

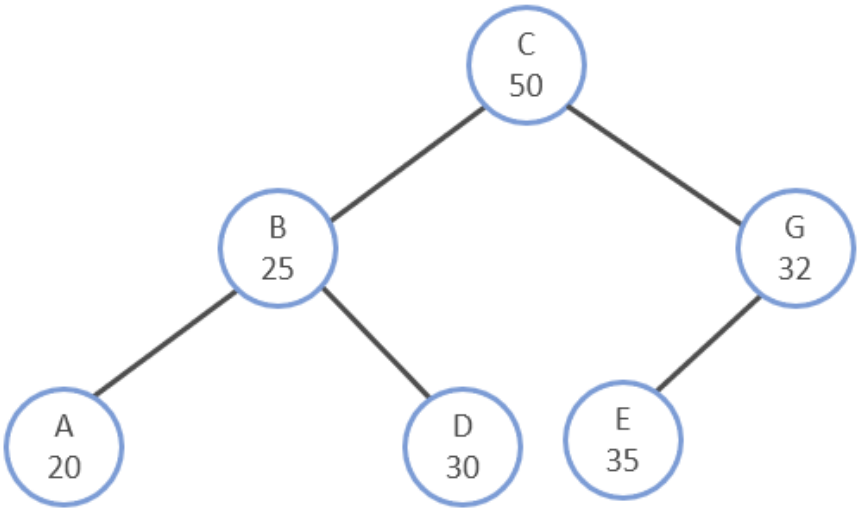
Question 20

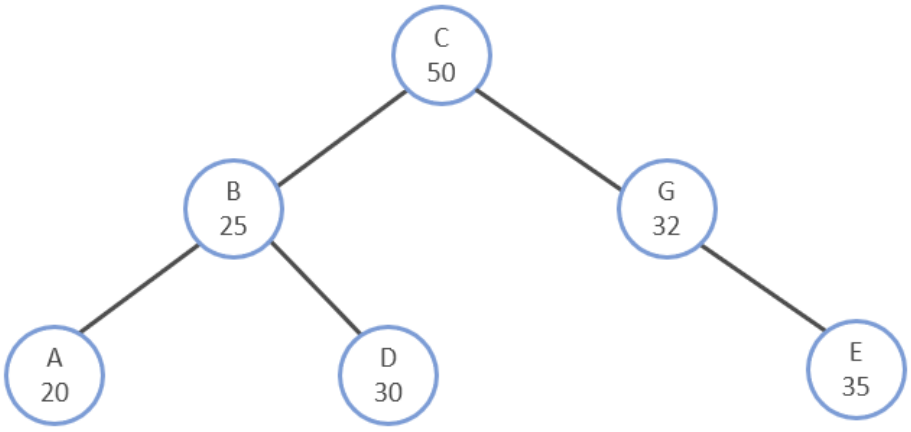
1 / 1 pts

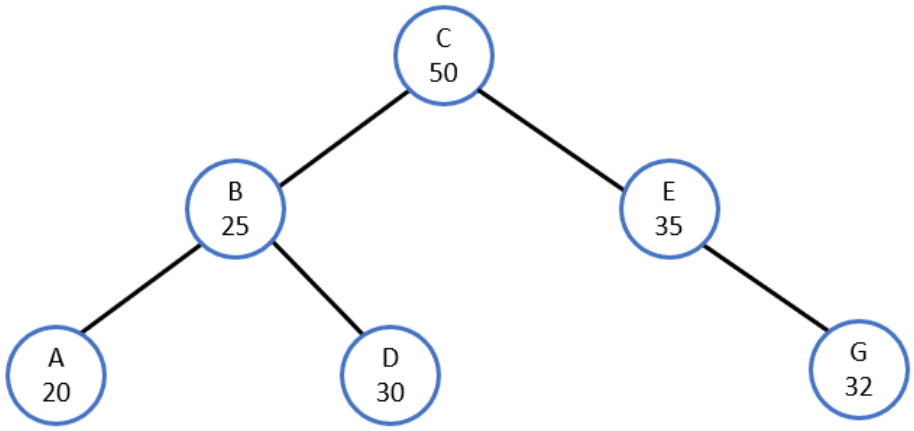
Identify the new treap created after deleting node F, 40.




C
50
B
25
E
35
A
20
D
30
G
32


C
50
B
25
G
32
A
20
D
30
E
35

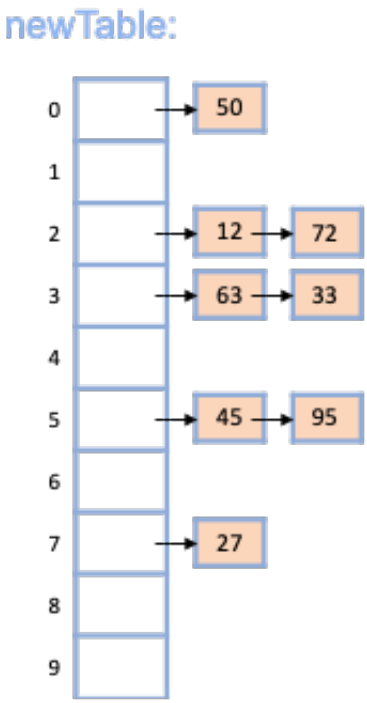

C
50
B
25
G
32
A
20
D
30
E
35



Question 21

1 / 1 pts

Consider the following hash table, and a hash function of key % 10.
How many list items will be compared for the search operations?
HashInsert(newTable, item 25)
HashInsert(newTable, item 54)
HashRemove(newTable, 27)
HashInsert(newTable, item 84)
HashInsert(newTable, item 83)
HashSearch(newTable, 72)
HashSearch(newTable, 77)
HashSearch(newTable, 63)

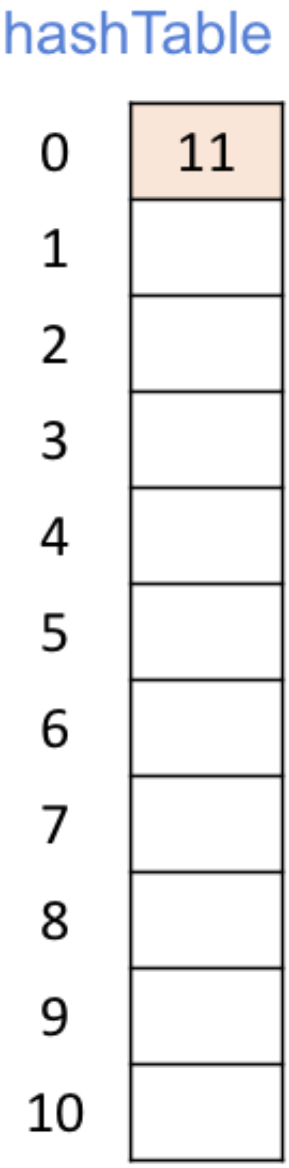


- 2; 0; 1
- 2; 0; 3
- 2; 0; 2
- 1; 0; 3

Question 22

1 / 1 pts

Given the following table, where a hash function returns key % 11 and quadratic probing is used with c1 = 1 and c2 = 1, which values can be inserted sequentially without collision?



- 22, 33, 44
- 22, 34, 45

- [x] 23, 35, 47
- [ ] 23, 34, 45

Question 23

1 / 1 pts

Using double hashing, how is the index determined when inserting item 20? hash1(key) = key % 11
hash2(key) = 5 - key % 5
and a hash table with a size of 10

- [ ] (20 % 11 + 1 * (5 - 20 % 5)) % 10
- [ ] (20 % 11 + 0 * (5 - 20 % 5)) % 11
- [ ] (20 % 11 + 1 * (5 - 20 % 5)) % 11
- [x] (20 % 11 + 0 * (5 - 20 % 5)) % 10

Question 24

1 / 1 pts

Which XXX completes the multiplicative string hash function?

```
HashMutliplicative(string key) {
   stringHash = InitialValue
   for (each character strChar in key) {
      XXX
   }
   return stringHash % N
}
```

- stringHash = (stringHash + HashMultiplier) * strChar
- stringHash = (stringHash * HashMultiplier) + strChar
- stringHash = (stringHash + HashMultiplier) + strChar
- stringHash = (stringHash * HashMultiplier) * strChar

Question 25

1 / 1 pts

Which XXX would replace the missing statement in the given SetDifference algorithm?

```
SetDifference(set1, set2) {
   result = Create new, empty set
   for each (element in set1) {
      if (XXX) {
         Add element to result
      }
   }
   return result
}
```

- SetSearch(set2, element⇢key) ≤ null
- SetSearch(set2, element⇢key) != null
- SetSearch(set2, element⇢key) == null
- SetSearch(set2, element⇢key) ≥ null

Question 26

1 / 1 pts

Which XXX should replace the missing statement in the selection sort algorithm given below?

```
SelectionSort(list, listSize) {
   for (ctr1 = 0; ctr1 < listSize - 1; ++ctr1) {
      mimIndex = ctr1
      for (ctr2 = ctr1 + 1; ctr2 < listSize; ++ctr2) {
         if (list[ctr2] < list[mimIndex]) {
            mimIndex = ctr2
         }
      }
      temp = list[ctr1]
      XXX
      list[mimIndex] = temp
   }
}
```

- list[ctr1] = list[mimIndex]
- list[ctr2] = list[mimIndex]
- list[mimIndex] = list[ctr1]
- list[mimIndex] = list[ctr2]

Question 27

1 / 1 pts

Which is the worst case runtime for a quicksort algorithm?

- $O(N^2)$
- $O(N \cdot log(N))$
- $O(N)$
- $O(2^N)$

Question 28

1 / 1 pts

Identify the correct MergeSort function.

-
```
MergeSort(myList, i, k) {
   j = 0
   if (i < k) {
      j = (i + k) / 2
      MergeSort(myList, i, j)
      MergeSort(myList, j + 1, k)
      Merge(myList, i, j, k)
   }
}
```

```
MergeSort(myList, i, k) {
   j = 0
   if (i > k) {
      j = (i + k) / 2
      MergeSort(myList, i, j)
      MergeSort(myList, j + 1, k)
      Merge(myList, i, j, k)
   }
}
```

```
MergeSort(myList, i, k) {
   j = 0
   if (i < k) {
      j = (i + k) / 2
      Merge(myList, i, j, k)
      MergeSort(myList, i, j)
      MergeSort(myList, j + 1, k)
   }
}
```

```
MergeSort(myList, i, k) {
   j = 0
   if (i < k) {
      j = (i + k) / 2
      MergeSort(myList, i, j)
      MergeSort(myList, j, k + 1)
      Merge(myList, i, j + 1, k)
   }
}
```

Question 29

1 / 1 pts

Which list cannot be sorted using the standard Radix sort algorithm?

- (45.1, -65.6, 89.8, -34.5, 23.3)
- (-67, -89, -34, -10, -65)

- (1, 22, 333, 4444, 55555)
- (44, 789, 5678, 90, 1)

Question 30

1 / 1 pts

Which XXX would replace the missing statement in the given bucket sort algorithm?

```
BucketSort(numbers, numbersSize, bucketCount) {
   if (numbersSize < 1)
      return
   buckets = Create list of bucketCount buckets
   maxValue = numbers[0]
   for (i = 1; i < numbersSize; i++) {
      if (numbers[i] > maxValue)
         maxValue = numbers[i]
   }
   for each (number in numbers) {
      XXX
      Append number to buckets[index]
   }
   for each (bucket in buckets)
      Sort(bucket)


      result = Concatenate all buckets together
   Copy result to numbers
}
```

- index = floor(number * bucketCount / (maxValue + 1))
- index = floor((bucketCount - 1) / maxValue)
- index = floor(number * (bucketCount +1) / maxValue)
- index = floor(number * bucketCount / maxValue)

Quiz Score: **30** out of 30