http://www.na.edu

E-mail: moodle@na.edu

NORTH AMERICAN UNIVERSITY
INSPIRATION INNOVATION GLOBAL COMPETENCE

Geraldo Braho

Started on	Thursday, 8 December 2016, 11:30 AM
State	Finished
Completed on	Thursday, 8 December 2016, 12:30 PM
Time taken	1 hour
Marks	47.00/50.00
Grade	**94.00** out of 100.00

Question 1

Correct Mark 1.00 out of 1.00

_____ is a way of arranging data to make solving a particular problem easier.

Select one:

- a. Algorithm
- b. Definition
- c. Programming Language
- d. Data Structure ✓

Your answer is correct.

The correct answer is: Data Structure

Question 2

Correct Mark 1.00 out of 1.00

Which algorithm calculates the result faster?

Select one:

a. def RaisetoPower(A,P):
```
result=1
for i in xrange(P):
    result*=A
return result
```

b. None of them!

c. def RaiseToPower(A,P):
```
#calculate A^1, A^2, A^4, A^8, and so on
#until you get to a value AN where N + 1 > P
i=1
result=1
while (P>=1):#
    if (P % 2)==1:
        #result = result * powerlist[i]
        result=result*(A)
        #print i

    P=P/2
    A=A*A
```

d. I am not sure :)

Your answer is correct.

The correct answer is: def RaiseToPower(A,P):
```
#calculate A^1, A^2, A^4, A^8, and so on
#until you get to a value AN where N + 1 > P
```

```
i=1
result=1
while (P>=1):#
    if (P % 2)==1:
        #result = result * powerlist[i]
        result=result*(A)
        #print i

    P=P/2
    A=A*A
```

Question 3

Incorrect Mark 0.00 out of 1.00

What does the following code do?

```
def aFunction():
    i=10
    while i>0:
        dice=random.randint(1,6)
        if dice <=3:
            print "heads"
        else:
            print "tails"
        i=i-1
```

Select one:

- a. Generates a fair 6-sided dice from a fair coin
- b. Generates a fair coin from a fair 6-sided dice
- c. Generates a fair coin from an unfair 6-sided dice
- d. Generates an unfair coin from a fair 6-sided dice ✗

Your answer is incorrect.

The correct answer is: Generates a fair coin from a fair 6-sided dice

Question 4

Correct Mark 1.00 out of 1.00

Which one of the following is true about this coin:

```
def aCoin():
    number = random.randint(1,4)
    if number == 3:
        return "tails"
    else:
        return "heads"
```

Select one:

- a. Fair coin, 50% tails
- b. Biased coin, 25% heads
- c. Biased coin, 50% tails
- d. Biased coin, 25% tails ✓

Your answer is correct.

The correct answer is: Biased coin, 25% tails

Question 5

Correct Mark 1.00 out of 1.00

Which one of the following can be used as pseudo random number generator?

Select one:

- a. Heat measurements of the CPU
- b. Linear congruential generator ✓
- c. Data structures
- d. Stock market prices

Your answer is correct.

The correct answer is: Linear congruential generator

Question 6

Correct Mark 1.00 out of 1.00

Lets say you have a sign up sheet for a lottery and you want to give all of the participants the same probability of winning. So you put their names into an array as explained in the book and shuffled that array. Do you think you need to shuffle the array several times to get a reasonably random result?

Select one:

- a. No ✓
- b. I don't know :)
- c. Depends on the array size
- d. Yes

Your answer is correct.

The correct answer is: No

Question 7

Correct Mark 1.00 out of 1.00

What is the usage of the following pseudo-code?

```
Function(Integer: number)
   List Of Integer: variable_list
   While (number Mod 2 == 0)
      variable_list.Add(2)
      number = number / 2
   End While
   Integer: i = 3
   Integer: max_variable = Sqrt(number)
   While (i <= max_variable)
      While (number Mod i == 0)
         variable_list.Add(i)
         number = number / i
         max_variable = Sqrt(number)
      End While
      i = i + 2
   End While
   If (number > 1) Then variable_list.Add(number)

   Return variable_list
End Function
```

Select one:

- a. Finding odd numbers in the list
- b. Finding even numbers in the list
- c. None of the above
- d. Finding prime factors (O(N) solution)
- e. Finding prime factors (O(sqrt(N) solution) ✓

The correct answer is: Finding prime factors (O(sqrt(N) solution)

Question 8

Correct Mark 1.00 out of 1.00

What is the usage of the following pseudo-code?

```
function(String: array[])
   Integer: max_i = <Upper bound of array>
   For i = 0 To max_i - 1
      Integer: j = <pseudorandom number between i and max_i inclusive>
      <Swap the values of array[i] and array[j]>
   Next i
End function
```

Select one:

- a. Finding min value of the array
- b. None of the above
- c. Finding max value of the array
- d. Sorting the array
- e. Randomizing the array ✓

The correct answer is: Randomizing the array

Question 9

Correct Mark 1.00 out of 1.00

What does the following Python Script do?

```
def function(node):
    if node.next == None: return True
    elif node.next.next == None: return True
    node = node.next
    while (node.next != None):
        if node.value > node.next.value:
            return False
        node = node.next
    return True
```

Select one:

- a. Checking linked list if it's sorted or not ✓
- b. Checking linked list to find min value
- c. Checking linked list's size
- d. None of them
- e. Checking linked list to find max value

Your answer is correct.

The correct answer is: Checking linked list if it's sorted or not

Question 10

Incorrect Mark 0.00 out of 1.00

Which one is an algorithm to add an item at the top of a doubly linked list.

Select one:

a. def Function(top, new_cell):
```
    # Update the next links
    new_cell.next = top.next
    top.next = new_cell

    # Update the prev links
    new_cell.prev = new_cell
    new_cell.next = top
```

b. def Function(top, new_cell):
```
    # Update the next links
    new_cell.prev = top.next
    top.prev = new_cell

    # Update the prev links
    new_cell.next.prev = new_cell
    new_cell.prev = top
```
✗

c. def Function(top, new_cell):
```
    # Update the next links
    new_cell.next = top.prev
    top.next = new_cell

    # Update the prev links
    new_cell.next.prev = new_cell
    new_cell.prev.next = top
```

d. def Function(top, new_cell):
```
    # Update the next links
    new_cell.next = top.next
    top.next = new_cell
```

```
# Update the prev links
new_cell.next.prev = new_cell
new_cell.prev = top
```

Your answer is incorrect.

The correct answer is: def Function(top, new_cell):

```
# Update the next links
new_cell.next = top.next
top.next = new_cell

# Update the prev links
new_cell.next.prev = new_cell
new_cell.prev = top
```

Question 11

Correct Mark 1.00 out of 1.00

If at some point in the game a player has 30 possible moves, the tree at that point has __ possible branches.

Select one:

- a. 30 ✓
- b. 15
- c. 2
- d. 60

Your answer is correct.

The correct answer is: 30

Question 12

Correct Mark 1.00 out of 1.00

Minimax is a game tree search strategy in which at each move you try to minimize the maximum value your opponent can achieve.

Select one:

- True ✓
- False

The correct answer is 'True'.

Question 13

Correct Mark 1.00 out of 1.00

Which technique is more effective when searching trees?

Select one:

- a. Exhaustive search
- b. Linear search
- c. Random search
- d. Branch and bound search ✓

Your answer is correct.

The correct answer is: Branch and bound search

Question 14

Correct Mark 1.00 out of 1.00

You can model games such as chess, checkers, Go, and tic-tac-toe with a game tree where each branch represents a
move by one of the players.

Select one:

- True ✓
- False

The correct answer is 'True'.

Question 15

Correct Mark 1.00 out of 1.00

Simulated annealing is an improved version of the simple improvement to the heuristic random search. Simulated annealing initially makes small changes to a solution and then over time makes larger and larger changes to try to improve the solution.

Select one:

- True
- False ✓

The correct answer is 'False'.

Question 16

Correct Mark 1.00 out of 1.00

What would be the output of the following code if quadratic probing policy is used?

```
table = [0]*10
def myhash(x):return x%10

def insert(table, value):
    table[myhash(value)]=value #This line is changed with quadratic probing policy

insert(table, 3)
insert(table, 4)
insert(table, 13)
print table
```

Select one:

a.

```
[0, 0, 0, 3, 4, 0, 0, 13, 0, 0]
```

✓

b.

```
[0, 0, 0, 3, 0, 0, 0, 0, 0, 0]
```

c.

```
[0, 0, 13, 3, 4, 0, 0, 0, 0, 0]
```

d.

```
[0, 0, 0, 13, 4, 0, 0, 0, 0, 0]
```

Your answer is correct.

The correct answer is:

```
[0, 0, 0, 3, 4, 0, 0, 13, 0, 0]
```

Question 17

Correct Mark 1.00 out of 1.00

Which one of the following is not a collision resolution policy?

Select one:

- a. Chaining
- b. Double Hashing
- c. Linear Hashing ✓
- d. Open Addressing

Your answer is correct.

The correct answer is: Linear Hashing

Question 18

Correct Mark 1.00 out of 1.00

Hash tables associate a key to a value, that is why they are sometimes called _____.

Select one:

- a. None of them
- b. chains
- c. arrays
- d. associative arrays ✓

Your answer is correct.

The correct answer is: associative arrays

Question 19

Correct Mark 1.00 out of 1.00

What would be the output of the following code?

```
table = [0]*10
def myhash(x):return x%10

def insert(table, value):
    table[myhash(value)]=value

insert(table, 3)
insert(table, 4)
insert(table, 13)
print table
```

Select one:

a.
```
[0, 0, 0, 13, 4, 0, 0, 0, 0, 0]
```
✓

b.
```
[0, 0, 0, 3, 4, 0, 0, 0, 0, 0]
```

c.
```
[0, 0, 0, 3, 0, 0, 0, 0, 0, 0]
```

d.
```
[0, 0, 0, 3, 4, 13, 0, 0, 0, 0]
```

Your answer is correct.

The correct answer is:

```
[0, 0, 0, 13, 4, 0, 0, 0, 0, 0]
```

Question 20

Correct Mark 1.00 out of 1.00

What would be the output of the following code if linear probing policy is used?

```
table = [0]*10
def myhash(x):return x%10

def insert(table, value):
    table[myhash(value)]=value #This line is changed with linear probing policy

insert(table, 3)
insert(table, 4)
insert(table, 13)
print table
```

Select one:

a.
```
[0, 0, 13, 3, 4, 0, 0, 0, 0, 0]
```

b.
```
[0, 0, 0, 13, 4, 0, 0, 0, 0, 0]
```

c.
```
[0, 0, 0, 3, 0, 0, 0, 0, 0, 0]
```

d.
```
[0, 0, 0, 3, 4, 13, 0, 0, 0, 0]
```

Your answer is correct.

The correct answer is:

```
[0, 0, 0, 3, 4, 13, 0, 0, 0, 0]
```

Comment:

Question 21

Correct Mark 1.00 out of 1.00

What would be the output of the following code if quadratic probing policy is used?

```
table = [0]*10
def myhash(x):return x%10

def insert(table, value):
    table[myhash(value)]=value #This line is changed with quadratic probing policy

insert(table, 3)
insert(table, 4)
insert(table, 13)
insert(table, 33)
print table
```

Select one:

a.

```
[0, 0, 33, 3, 4, 0, 0, 13, 0, 0]
```

b.

```
[0, 0, 0, 3, 4, 13, 33, 0, 0, 0]
```

c.

```
[0, 0, 0, 3, 4, 0, 0, 13, 0, 33]
```

d.

```
[0, 0, 0, 3, 0, 0, 0, 0, 0, 0]
```

Your answer is correct.

The correct answer is:

```
[0, 0, 33, 3, 4, 0, 0, 13, 0, 0]
```

Question 22

Correct Mark 1.00 out of 1.00

_____ Algorithm is best suited to find the shortest path between nodes in a network.

Select one:

a. Exhaustive

b. Minimax

c. Dijkstra's ✓

d. Heuristic

Your answer is correct.

The correct answer is: Dijkstra's

Question 23

Correct Mark 1.00 out of 1.00

Degree of a network is determined by _____ degree of any of its nodes.

Select one:

- a. largest ✓
- b. median
- c. None of the above
- d. smallest

Your answer is correct.

The correct answer is: largest

Question 24

Correct Mark 1.00 out of 1.00

Depth-first traversal algorithm implemented for trees might have a problem on networks because of _____.

Select one:

- a. nodes
- b. links
- c. missing root
- d. cycles ✓

Your answer is correct.

The correct answer is: cycles

Question 25

Correct Mark 1.00 out of 1.00

Which one of the following functions have a correct base case?

Select one:

- a. def Factorial(n):
  if (n == 0): return n-1
  return n * Factorial (n - 1)
- b. def Factorial(n):
  if (n == 0): return n
  return n * Factorial (n - 1)
- c. def Factorial(n):
  if (n > 0): return 1
  return n * Factorial (n - 1)
- d. def Factorial(n):
  if (n == 0): return 1
  return n * Factorial (n - 1) ✓

Your answer is correct.

The correct answer is: def Factorial(n):
if (n == 0): return 1
return n * Factorial (n - 1)

Question 26

Correct Mark 1.00 out of 1.00

Which one of the following is not naturally recursive?

Select one:

- a. Tower of Hanoi
- b. Factorial
- c. Fibonacci
- d. Sum of N numbers ✓

Your answer is correct.

The correct answer is: Sum of N numbers

Question 27

Correct Mark 1.00 out of 1.00

Which one of the following functions is a correct recursive function?

Select one:

a. def Fibonacci(n):
    if (n <= 1): return n
    return Fibonacci(n - 1) + Fibonacci(n)

b. def Factorial(n):
    if (n == 0): return 1
    return n * Factorial (n - 1) ✓

c. def factor(n):
    num = 1
    while n >= 1:
        num = num * n
        n = n - 1
    return num

d. def Factorial(n):
    return n * Factorial (n - 1)

Your answer is correct.

The correct answer is: def Factorial(n):
    if (n == 0): return 1
    return n * Factorial (n - 1)

Question 28

Correct Mark 1.00 out of 1.00

Recursion is not always the best solution.

Select one:

- True ✓
- False

The correct answer is 'True'.

Question 29

Correct Mark 1.00 out of 1.00

Which one of the following is a recursive implementation of binary search?

Select one:

- a.
```
def BinarySearchRecursive(values,target):
    #base case
    if (len(values) == 0): return False


    #recursion
    mid = len(values) // 2
    if (target == values[mid]):
        return True
    if (target > values[mid]):
        return BinarySearchRecursive(values[:mid],target)
    elif (target < values[mid]):
        return BinarySearchRecursive(values[mid+1:],target)
```
- b.
```
def BinarySearchRecursive(values,target):
    #base case
```

```
if (len(values) == 0): return False


#recursion
mid = len(values) // 2
if (target == values[mid]):
    return True
if (target < values[mid]):
    return BinarySearchRecursive(values[:mid],target)
elif (target > values[mid]):
    return BinarySearchRecursive(values[mid+1:],target)
```

c.
```
def BinarySearchRecursive(values,target):
#base case
if (len(values) > 0): return True


#recursion
mid = len(values) // 2
if (target == values[mid]):
    return True
if (target < values[mid]):
    return BinarySearchRecursive(values[:mid],target)
elif (target > values[mid]):
    return BinarySearchRecursive(values[mid+1:],target)
```

d.
```
def BinarySearchRecursive(values,target):
#base case
if (len(values) == 0): return False


#recursion
if (target == values[mid]):
    return True
if (target < values[mid]):
```

```
        return BinarySearchRecursive(values[:mid],target)
    elif (target > values[mid]):
        return BinarySearchRecursive(values[mid+1:],target)
```

Your answer is correct.

The correct answer is: def BinarySearchRecursive(values,target):

```
    #base case
    if (len(values) == 0): return False



    #recursion
    mid = len(values) // 2
    if (target == values[mid]):
        return True
    if (target < values[mid]):
        return BinarySearchRecursive(values[:mid],target)
    elif (target > values[mid]):
        return BinarySearchRecursive(values[mid+1:],target)
```

Question 30

Correct Mark 1.00 out of 1.00

What is the complexity of Linear Search?

Select one:

- a. O(N*N)
- b. O(log (log N))
- c. O(N) ✓
- d. O(log N)

Your answer is correct.

The correct answer is: O(N)

Question 31

Correct Mark 1.00 out of 1.00

What is the complexity of Interpolation Search?

Select one:

- a. O(log (log N)) ✓
- b. O(log N)
- c. O(N*N)
- d. O(N)

Your answer is correct.

The correct answer is: O(log (log N))

Question 32

Correct Mark 1.00 out of 1.00

Which one is the correct interpolation search?

Select one:

a. def InterpolationSearch(values, target):
```
min = 0
max = len(values) - 1
while (min <= max):
# Find the dividing item.
   scale = (target - min)/(max - min)
   mid = min + (max - min) * scale
   if (values[mid] == target): return mid
   if (values[mid]>target):
      min = mid + 1
   else:
      max = mid - 1
return -1
```

b. def InterpolationSearch(values, target):
```
min = 0
max = len(values) - 1
while (min <= max):
# Find the dividing item.
   scale = (target - values[min])/(values[max] - values[min])
   mid = (max - min) * scale
   if (values[mid] == target): return mid
   if (values[mid]>target):
      min = mid - 1
   else:
      max = mid + 1
return -1
```

c. def InterpolationSearch(values, target):
```
min = 0
max = len(values) - 1
while (min <= max):
```

```
    # Find the dividing item.
      scale = (target - values[min])/(values[max] - values[min])
      mid = min + (max - min) * scale
      if (values[mid] == target): return mid
      if (values[mid]>target):
        min = mid + 1
      else:
        max = mid - 1
    return -1
```

✓

d.
```
def InterpolationSearch(values, target):
    min = 0
    max = len(values) - 1
    while (min <= max):
    # Find the dividing item.
      scale = (target - values[min])/(values[max] - values[min])
      mid = (max - min) * scale
      if (values[mid] == target): return mid
      if (values[mid]>target):
        min = mid + 1
      else:
        max = mid - 1
    return -1
```

Your answer is correct.

The correct answer is:
```
def InterpolationSearch(values, target):
  min = 0
  max = len(values) - 1
  while (min <= max):
  # Find the dividing item.
    scale = (target - values[min])/(values[max] - values[min])
    mid = min + (max - min) * scale
    if (values[mid] == target): return mid
```

```
    if (values[mid]>target):
        min = mid + 1
    else:
        max = mid - 1
return -1
```

Question 33

Correct Mark 1.00 out of 1.00

How can you fix following binary search?

```
def BinarySearch(values,target):
    min = 0
    max = len(values) - 1
    while (min <= max):
    # Find the dividing item.
        [missing code]
        # See if we need to search the left or right half.
        if (target < values[mid]):
            max = mid - 1
        elif (target > values[mid]):
            min = mid + 1
        else: return mid

# If we get here, the target is not in the array.
    return -1
```

Select one:

- a. mid = max / 2
- b. max = (min + max) / 2
- c. min = (min + max) / 2
- d. mid = (min + max) / 2 ✓

Your answer is correct.

The correct answer is: mid = (min + max) / 2

Question 34

Correct Mark 1.00 out of 1.00

What is the complexity of Binary Search?

Select one:

- a. O(log (log N))
- b. O(N*N)
- c. O(N)
- d. O(log N) ✓

Your answer is correct.

The correct answer is: O(log N)

Question 35

Correct Mark 1.00 out of 1.00

Which one is not an $O(N^2)$ algorithm?

Select one:

- a. Selection Sort
- b. Merge Sort ✓
- c. Insertion Sort
- d. Bubble Sort

Your answer is correct.

The correct answer is: Merge Sort

Question 36

Correct Mark 1.00 out of 1.00

What is the name of the following sorting algorithm?

```
def Function(list):
    # This will be our loop condition, loop until False
    not_sorted = True
    while not_sorted:
        # Assuming we won't find a pair to swap
        not_sorted = False
        # Search list for the adjacent items that are out of order
        for i in range(len(list)-1):
            if list[i+1] < list[i]:
                # Swap if true
                list[i+1], list[i] = list[i], list[i+1]
                # When we swap assign True again for another loop
                not_sorted = True
                print list
        if not_sorted: print 'here goes the next round'
    return list
```

Select one:

- a. Selection Sort
- b. Merge Sort
- c. Bubble Sort ✓
- d. Insertion Sort

Your answer is correct.

The correct answer is: Bubble Sort

Question 37

Correct Mark 1.00 out of 1.00

Which algorithm work best for the following input?
100,000 integers with values between 0 and 10 million

Select one:

- a. Count Sort
- b. Bucket Sort ✓
- c. Merge Sort
- d. Selection Sort

Your answer is correct.

The correct answer is: Bucket Sort

Question 38

Correct Mark 1.00 out of 1.00

Which one is a sub O(N log N) algorithm?

Select one:

- a. Heap Sort
- b. Quick Sort
- c. Merge Sort
- d. Bucket Sort ✓

Your answer is correct.

The correct answer is: Bucket Sort

Question 39

Correct Mark 1.00 out of 1.00

Bucket Sort would work efficiently if the input has a nonuniform distribution.

Select one:

- True
- False ✓

The correct answer is 'False'.

Question 40

Correct Mark 1.00 out of 1.00

Which algorithm work best for the following input?
100,000 integers with values between 0 and 1,000

Select one:

- a. Counting Sort ✓
- b. Insertion Sort
- c. Merge Sort
- d. Selection Sort

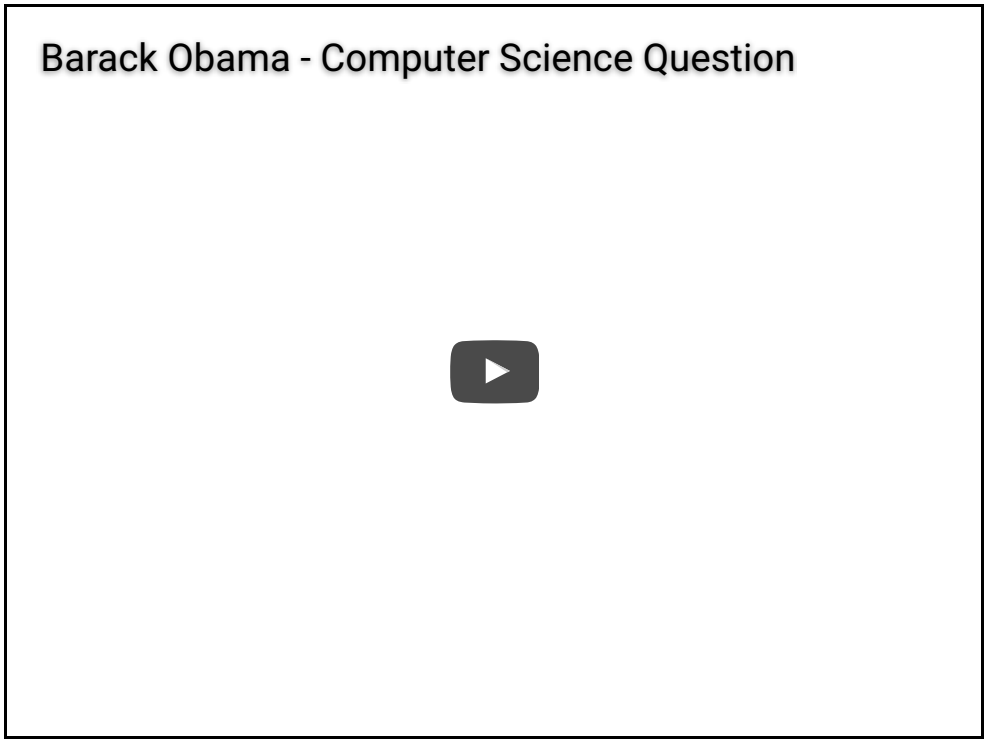
Your answer is correct.

The correct answer is: Counting Sort

Question 41

Correct Mark 1.00 out of 1.00

What is the most efficient way to sort a million 32-bit integers?



Select one:

- a. Merge Sort
- b. Quick Sort
- c. Heap Sort
- d. Radix Sort ✓

Your answer is correct.

The correct answer is: Radix Sort

Question 42

Correct Mark 1.00 out of 1.00

Having the same height, which one of the tree types might have more leaves?

Select one:

- a. Sequoia Tree
- b. Complete Tree
- c. Full Tree
- d. Perfect Tree ✓

Your answer is correct.

The correct answer is: Perfect Tree

Question 43

Correct Mark 1.00 out of 1.00

A binary tree with the height of 5 has a degree of _____.

Select one:

- a. 4
- b. 5
- c. 3
- d. 2 ✓

Your answer is correct.

The correct answer is: 2

Question 44

Correct Mark 1.00 out of 1.00

Which one of the following is true about the height of the root?

Select one:

- a. it changes with the height of the tree ✓
- b. zero
- c. it changes with the degree of the tree
- d. one

Your answer is correct.

The correct answer is: it changes with the height of the tree

Question 45

Correct Mark 1.00 out of 1.00

A node that has no parent is called _____.

Select one:

- a. leaf
- b. root ✓
- c. parent node
- d. external node

Your answer is correct.

The correct answer is: root

Question 46

Correct Mark 1.00 out of 1.00

Trees are highly recursive data structures for hierarchical data. Which one is not a good example for this type of data:

Select one:

- a. Dictionary data ✓
- b. Organizational charts
- c. Graphs
- d. Parts of a car

Your answer is correct.

The correct answer is: Dictionary data

Question 47

Correct Mark 1.00 out of 1.00

What would we like to see in an algorithm?

Select one:

- a. Correctness
- b. All of them ✓
- c. Maintainability
- d. Efficiency

Your answer is correct.

The correct answer is: All of them

Question 48

Correct Mark 1.00 out of 1.00

What is the complexity of finding the minimum of a list?

Select one:

- a. O(N) ✓
- b. O(1)
- c. O(2)
- d. $O(N^2)$

Your answer is correct.

The correct answer is: O(N)

Question 49

Correct Mark 1.00 out of 1.00

What is the complexity of printing the first element of a list?

Select one:

- a. O(1) ✓
- b. $O(N^2)$
- c. O(2)
- d. O(N)

Your answer is correct.

The correct answer is: O(1)

Question 50

Incorrect Mark 0.00 out of 1.00

What is the complexity of the following algorithm?

```
for (int count = 1; count < n; count++)
 for (int count2 = 1; count2 < n; count2 = count2 * 2)
 {
    // some sequence of O(1) steps
 }
```

Select one:

- a. O(2)
- b. O(N log N)
- c. O(1)
- d. $O(N^2)$ ✗

Your answer is incorrect.

The correct answer is: O(N log N)