http://www.na.edu

E-mail: moodle@na.edu

NORTH AMERICAN UNIVERSITY
INSPIRATION INNOVATION GLOBAL COMPETENCE

Geraldo Braho

Started on	Thursday, 13 October 2016, 1:02 PM
State	Finished
Completed on	Thursday, 13 October 2016, 1:31 PM
Time taken	29 mins 42 secs
Marks	24.00/25.00
Grade	**96.00** out of 100.00

Question 1

Correct Mark 1.00 out of 1.00

What is the complexity of the following algorithm?

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

Select one:

- a. $O(4N^2)$
- b. $O(1)$
- c. $O(N)$
- d. $O(N^2)$ ✓

Your answer is correct.

The correct answer is: $O(N^2)$

Question 2

Correct Mark 1.00 out of 1.00

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

Select one:

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

Your answer is correct.

The correct answer is: Linear congruential generator

Question 3

Correct Mark 1.00 out of 1.00

Which algorithm calculates the result faster?

Select one:

- a.
```
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
```

- b. None of them!
- c. I am not sure :)
- d. def RaisetoPower(A,P):

```
    result=1
    for i in xrange(P):
        result*=A
    return result
```

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 4

Correct Mark 1.00 out of 1.00

What is this algorithm?

```
def surprise(A,B):
    while (B != 0):
        remainder = A % B
        A = B
        B = remainder
    return A
```

Select one:

- a. Lowest Common Multiplier
- b. Greatest Common Divisor ✓
- c. Remainder
- d. Exponentiation

Your answer is correct.

The correct answer is: Greatest Common Divisor

Question 5

Correct Mark 1.00 out of 1.00

What is the running time of the following algorithm?

```
def surprise(A,B):
    while (B != 0):
        remainder = A % B
        A = B
        B = remainder
    return A
```

Select one:

- a. O(log N) ✓
- b. O(A%B)
- c. O(1)
- d. O(N)

Your answer is correct.

The correct answer is: O(log N)

Question 6

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. None of them
- b. Checking linked list's size
- c. Checking linked list if it's sorted or not ✓
- d. Checking linked list to find min value
- 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 7

Correct Mark 1.00 out of 1.00

What is the following sample pseudo-code's complexity?

```
list.add(2)
for (i = 3 to n)
   isPrime=true
   for each (j in list)
      if j > sqrt(i)
         break
      if ((i mod j) = 0)
         isPrime=false
         break
   if (isPrime)
      list.add(i)
return list
```

Select one:

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

The correct answer is: O(N * sqrt(N))

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. Randomizing the array ✓
- d. Sorting the array
- e. Finding max value of the array

The correct answer is: Randomizing the array

Question 9

Correct Mark 1.00 out of 1.00

Which one of pseudo-codes is the right choice for adding cell at the end of the linked list?

Select one:

- a. Function(Cell: top, Cell: new_cell)
  new_cell.Next = top.Next

```
    top.Next = new_cell
End Function
```

b. Function(Cell: top, Value: target)

```
    While (top.Next != null)
        If (top.Next.Value == target) Then Return top
        top = top.Next
    End While
    Return null
End Function
```

c. Function(Cell: top, Value: target)

```
    While (top != null)
        If (top.Value == target) Then Return top
        top = top.Next
    End While
    Return null
End Function
```

d. Function(Cell: top, Cell: new_cell)

```
    While (top.Next != null)
        top = top.Next
    End While
    top.Next = new_cell
    new_cell.Next = null
End Function
```

e. Function(Cell: top)

```
    While (top != null)
        Print top.Value
        top = top.Next
    End While
End Function
```

Your answer is correct.

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

```
  While (top.Next != null)
    top = top.Next
  End While
  top.Next = new_cell
  new_cell.Next = null
End Function
```

Question 10

Correct Mark 1.00 out of 1.00

Which one of the following code is for calculating sample variance m2?

$$m_2 \equiv \frac{1}{N}\sum_{i=1}^{N}(x_i - m)^2,$$

where $m = \bar{x}$ the sample mean (average) and N is the sample size.

Select one:

a.
```
def Function(array):
    total = sum(array)
    average = total / len(array)

    #Find the sample variance.
    sum_of_squares = 0
    for i in range(len(array)):
        sum_of_squares =  (array[i] - average)**2
    return sum_of_squares / average
```

b.
```
def Function(array):
    total = sum(array)
    average = total / len(array)

    #Find the sample variance.
    sum_of_squares = 0
```

```
for i in range(len(array)):
    sum_of_squares = sum_of_squares + (array[i] - average)
return sum_of_squares / len(array)
```

c.
```
def Function(array):
    total = sum(array)
    average = 0

    #Find the sample variance.
    sum_of_squares = total
    for i in range(len(array)):
        sum_of_squares = sum_of_squares + (array[i] - average)
    return sum_of_squares / len(array)
```

d.
```
def Function(array):
    total = sum(array)
    average = total / len(array)

    #Find the sample variance.
    sum_of_squares = 0
    for i in range(len(array)):
        sum_of_squares = sum_of_squares + (array[i] - average)**2
    return sum_of_squares / len(array)
```
✓

Your answer is correct.

The correct answer is:
```
def Function(array):
    total = sum(array)
    average = total / len(array)

    #Find the sample variance.
    sum_of_squares = 0
    for i in range(len(array)):
        sum_of_squares = sum_of_squares + (array[i] - average)**2
    return sum_of_squares / len(array)
```

Question 11

Correct Mark 1.00 out of 1.00

Which one of the following code can be used to find the minimum element of an array?

Select one:

a. def Function (array):
```
total = 0
for i in range(len(array)):
    total += array[i]
return total / len(array)
```

b. def Function (array):
```
a = array[0]
for i in range(len(array)):
    if (array[i] < a):
        a = array[i]
return a
```
✓

c. def Function (array, target):
```
for i in range(len(array)):
    if (array[i] == target):
        return i
# the target isn't in the array
return -1
```

d. def Function (array, target):
```
for i in array:
    if (i == target):
        return i
# the target isn't in the array
return false
```

e. def Function (array):
```
a = array[0]
```

```
for i in range(len(array)):
    if (array[i] > a):
        a = array[i]
return a
```

Your answer is correct.

The correct answer is: def Function (array):

```
a = array[0]
for i in range(len(array)):
    if (array[i] < a):
        a = array[i]
return a
```

Question 12

Correct Mark 1.00 out of 1.00

Which one of the following code is a linear search and returns the target element?

Select one:

a. def Function (array, target):

```
for i in array:
    if (i == target):
        return i
# the target isn't in the array
return false
```

b. def Function (array):

```
total = 0
for i in range(len(array)):
    total += array[i]
return total / len(array)
```

c. def Function (array):

```
a = array[0]
for i in range(len(array)):
    if (array[i] < a):
        a = array[i]
return a
```

d. def Function (array, target):

```
for i in range(len(array)):
    if (array[i] == target):
        return i
# the target isn't in the array
return -1
```

e. def Function (array):

```
a = array[0]
for i in range(len(array)):
    if (array[i] > a):
        a = array[i]
return a
```

Your answer is correct.

The correct answer is: def Function (array, target):

```
for i in array:
    if (i == target):
        return i
# the target isn't in the array
return false
```

Question 13

Correct Mark 1.00 out of 1.00

Regular matrix multiplication is O(N^3).

Select one:

- True ✓
- False

The correct answer is 'True'.

Question 14

Correct Mark 1.00 out of 1.00

Consider an airline connectivity matrix that holds 1 in the [i, j] entry to indicate that there is a flight between city i and city j. The airline might have only 600 flights connecting 200 cities. In that case there would be only 600 nonzero values in an array of 40,000 entries (200x200). Even if the flights are symmetrical (for every i – j flight there is a j – i flight) and you store the connections in a special array which holds the half of the matrix, the array would hold only 300 nonzero entries out of a total of 20,100 entries. The array would be almost 99% unused. Which of the following would be a good solution?

Select one:

- a. Sparse Arrays ✓
- b. Rectangular
- c. Linked Arrays
- d. Triangular

Your answer is correct.

The correct answer is: Sparse Arrays

Question 15

Correct Mark 1.00 out of 1.00

_____ is a data structure where items are added and removed in first-in-first-out order.

Select one:

- a. Linked List
- b. Queue ✓
- c. Stack
- d. Array

Your answer is correct.

The correct answer is: Queue

Question 16

Correct Mark 1.00 out of 1.00

_____ is a data structure where items are added and removed in last-in-first-out order.

Select one:

- a. Array
- b. Queue
- c. Linked List
- d. Stack ✓

Your answer is correct.

The correct answer is: Stack

Question 17

Correct Mark 1.00 out of 1.00

Which one is not an O(N log N) algorithm?

Select one:

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

Your answer is correct.

The correct answer is: Counting Sort

Question 18

Correct Mark 1.00 out of 1.00

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

Select one:

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

Your answer is correct.

The correct answer is: Bucket Sort

Question 19

Correct Mark 1.00 out of 1.00

What is the name of the following sorting algorithm?

```
def Function(list):
    # Loop the number of elements in the list
    for i in xrange(1,len(list)):
        # save the value to be positioned
        value = list[i]
        # Find the position where value fits
        # in the ordered part of the list
        pos = i
        # Checking conditions
        while pos > 0 and value < list[pos - 1]:
            # shift the items during the search
            list[pos] = list[pos - 1]
            pos -= 1
        # Add it to empty space
        list[pos] = value
    return list
```

Select one:

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

Your answer is correct.

The correct answer is: Insertion Sort

Question 20

Correct Mark 1.00 out of 1.00

What is the name of the following sorting algorithm?

```
def Function(list):
    for i in range(len(list)):
        # Initialize the smallest element with index i (=0)
        smallest_index = i
        for j in range(i+1, len(list)):
            if list[j] < list[smallest_index]:
                # if jth element is smaller then smallest
                # change smallest to j
                smallest_index = j

        if smallest_index != i:
            # Swap the values
            list[i],list[smallest_index] = list[smallest_index],list[i]
    return list
```

Select one:

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

Your answer is correct.

The correct answer is: Selection Sort

Question 21

Correct Mark 1.00 out of 1.00

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

Select one:

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

Your answer is correct.

The correct answer is: Merge Sort

Question 22

Correct Mark 1.00 out of 1.00

What would we like to see in an algorithm?

Select one:

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

Your answer is correct.

The correct answer is: All of them

Question 23

Correct Mark 1.00 out of 1.00

What is the complexity of the following algorithm?

```
while (count <= n)
{
   count = count *2;
   // some sequence of O(1) steps
}
```

Select one:

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

Your answer is correct.

The correct answer is: O(log N)

Question 24

Correct Mark 1.00 out of 1.00

What is the complexity of the following algorithm?

```
while (count <= n)
{
   count = count *3;
   // some sequence of O(1) steps
}
```

Select one:

- a. O(log N/3)
- b. O(N)
- c. $O(N^2)$
- d. O(log N) ✓

Your answer is correct.

The correct answer is: O(log N)

Question 25

Incorrect Mark 0.00 out of 1.00

Which one of the following pseudocode can be used to find Greatest Common Divisor of two integers?

Select one:

- a.
```
Int GCD(a,b)
  While (b!=0)
    remainder = a mod b
    a=b
```

b=remainder
End While
Return a

b. Int GCD(a,b)
While (b!=0)
remainder = a mod b
b=remainder
End While
Return a ✗

c. Int GCD(a,b)
While (b!=0)
remainder = a mod b
a=b
b=remainder
End While
Return b

d. Int GCD(a,b)
While (b!=0)
remainder = a mod b
a=b
End While
Return a

Your answer is incorrect.

The correct answer is: Int GCD(a,b)
While (b!=0)
remainder = a mod b
a=b
b=remainder
End While
Return a