Quiz

Due Dec 13 at 23:59 **Points** 5 **Questions** 25 **Available** Dec 6 at 10:00 - Dec 13 at 23:59 8 days
Time Limit 120 Minutes

Instructions

The following Quiz is based on the following topics :

1. Asymptotic Notations
2. Time Complexity
3. Stack and Queue
4. Binary Tree
5. Heap

Regard's

I/C

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	57 minutes	5 out of 5

ⓘ Correct answers will be available Dec 15 at 23:59 - Dec 16 at 23:59.

Score for this quiz: **5** out of 5
Submitted Dec 7 at 20:21
This attempt took 57 minutes.

Question 1

0.2 / 0.2 pts

What is the time complexity of following code:

```
int a = 0, b = 0;
for (i = 0; i < N; i++) {
a = a + rand();
}
for (j = 0; j < M; j++) {
b = b + rand();
}
```

- O(N * M) time
- O(N + M) time
- O(nlogn)
- O(n+nlogn)

Question 2

0.2 / 0.2 pts

What is the time complexity of following code:

```
int a = 0;
```

```
for (i = 0; i < N; i++) {
for (j = N; j > i; j--) {
a = a + i + j;
}
}
```

- O(N*log(N))
- O(N * Sqrt(N))
- O(N)
- O(N*N)

Question 3

0.2 / 0.2 pts

What is the time complexity of following code:

```
int i, j, k = 0;
for (i = n / 2; i <= n; i++) {
for (j = 2; j <= n; j = j * 2) {
k = k + n / 2;
}
}
```

- (●) O(nLogn)
- ( ) O(n^2Logn)
- ( ) O(n)
- ( ) O(n^2)

Question 4

0.2 / 0.2 pts

What is the time complexity of following code:

```
int a = 0, i = N;
while (i > 0) {
a += i;
i /= 2;
}
```

- (●) O(log N)
- ( ) O(Sqrt(N))
- ( ) O(N)
- ( ) O(N / 2)

Question 5

0.2 / 0.2 pts

Find the complexity of the below program

```
function(int n)
{
if (n==1)
return;
for (int i=1; i<=n; i++)
{
for (int j=1; j<=n; j++)
{
printf("*");
break;
}
}
}
```

- O(n^2)
- o(logn)
- O(nlogn)
- (•) O(n)

Question 6

0.2 / 0.2 pts

Find the complexity of the below program

```
void function(int n)
{
int count = 0;
for (int i=n/2; i<=n; i++)
for (int j=1; j<=n; j = 2 * j)
for (int k=1; k<=n; k = k * 2)
count++;
}
```

- (•) O(n log2n)
- ( ) O(nlogn)
- ( ) O(M*N)
- ( ) O(n^2)

Question 7

0.2 / 0.2 pts

Find the complexity of the below program

```
void function(int n)
{
int i = 1, s =1;
while (s <= n)
{
i++;
s += i;
printf("*");
}
}
```

- O(n^2)
- O(√n)
- O(n)
- O(nlogn)

Question 8

0.2 / 0.2 pts

Find a tight upper bound on the complexity of the below program

```
void function(int n)
{
int count = 0;
```

```
for (int i=0; i<n; i++)
for (int j=i; j< i*i; j++)
if (j%i == 0)
{
for (int k=0; k<j; k++)
printf("*");
}
}
```

- O(n^3)
- O(n^2)
- O(n)
- (•) O(n^5)

Question 9

0.2 / 0.2 pts

Which one of the following is an application of Stack Data Structure?

- The stock span problem
- (•) All of the above

- Managing function calls
- Arithmetic expression evaluation

Question 10

0.2 / 0.2 pts

Following is an incorrect pseudocode for the algorithm which is supposed to determine whether a sequence of parentheses is balanced:

```
declare a character stack
while ( more input is available)
{
read a character
if ( the character is a '(' )
push it on the stack
else if ( the character is a ')' and the stack is not empty )
pop a character off the stack
else
print "unbalanced" and exit
}
print "balanced"
```

Which of these unbalanced sequences does the above code think is balanced?

- ())(()

- ( ) (( ) ( )))
- ( ) ( )) ( )
- (•) (( ))

Question 11

0.2 / 0.2 pts

The following postfix expression with single-digit operands is evaluated using a stack:

```
8 2 3 ^ / 2 3 * + 5 1 * -
Note that ^ is the exponentiation operator. The top two elements of the stack after the first * is evalua
ted are:
```

- ( ) 5,7
- (•) 6,1
- ( ) 1,5
- ( ) 3,2

Question 12

0.2 / 0.2 pts

Assume that the operators +, -, × are left-associative and ^ is right-associative. The order of precedence (from highest to lowest) is ^, x, +, -. The postfix expression corresponding to the infix expression a + b × c - d ^ e ^ f is

- (●) abc × + def ^ ^ -
- ( ) ab + c × d - e ^ f ^
- ( ) abc × + de ^ f ^ -
- ( ) - + a × bc ^ ^ def

Question 13

0.2 / 0.2 pts

The result evaluating the postfix expression 10 5 + 60 6 / * 8 – is

- ( ) 71
- (●) 142
- ( ) 284
- ( ) 213

Question 14

0.2 / 0.2 pts

Which of the following permutation can be obtained in the same order using a stack assuming that input is the sequence 5, 6, 7, 8, 9 in that order?

- 9, 8, 7, 5, 6
- 7, 8, 9, 5, 6
- 5, 9, 6, 7, 8
- (selected) 7, 8, 9, 6, 5

Question 15

0.2 / 0.2 pts

The best data structure to check whether an arithmetic expression has balanced parenthesis is a

- (selected) Stack
- List
- Tree
- Queue

Question 16

0.2 / 0.2 pts

The five items: A, B, C, D, and E are pushed in a stack, one after another starting from A. The stack is popped four items and each element is inserted in a queue. The two elements are deleted from the queue and pushed back on the stack. Now one item is popped from the stack. The popped item is

- (●) D
- ( ) B
- ( ) C
- ( ) A

Question 17

0.2 / 0.2 pts

Assume that the operators +, -, × are left-associative and ^ is right-associative. The order of precedence (from highest to lowest) is ^, x, +, -. The postfix expression corresponding to the infix expression a + b × c - d ^ e ^ f is

- ( ) - + a × bc ^ ^ def

- (●) abc × + def ^ ^ -
- ( ) ab + c × d - e ^ f ^
- ( ) abc × + de ^ f ^ -

Question 18

0.2 / 0.2 pts

Convert the following infix expression into its equivalent post fix expression (A + B^ D) / (E – F) + G

- ( ) ABD^ + EF / – G+
- (●) ABD^ + EF – / G+
- ( ) ABD + ^EF / – G+
- ( ) ABD^ + EF / – G+

Question 19

0.2 / 0.2 pts

A priority queue is implemented as a max-heap. Initially, it has five elements. The level-order traversal of the heap is as follows: 20, 18, 15, 13, 12 Two new elements ‘10’ and ‘17’ are inserted in

the heap in that order. The level-order traversal of the heap after the insertion of the element is:

- 20, 18, 17, 12, 13, 10, 15
- 20, 18, 17, 13, 12, 10, 15
- 20, 18, 17, 10, 12, 13, 15
- 20, 18, 17, 15, 13, 12, 10

Question 20

0.2 / 0.2 pts

Consider a standard Circular Queue 'q' implementation (which has the same condition for Queue Full and Queue Empty) whose size is 11 and the elements of the queue are q[0], q[1], q[2].....,q[10]. The front and rear pointers are initialized to point at q[2] . In which position will the ninth element be added?

- q[0]
- q[1]
- q[9]
- q[10]

Question 21

0.2 / 0.2 pts

A queue is implemented using an array such that ENQUEUE and DEQUEUE operations are performed efficiently. Which one of the following statements is CORRECT (n refers to the number of items in the queue)?

- (●) Both operations can be performed in O(1) time
- ( ) At most one operation can be performed in O(1) time but the worst case time for the other operation will be Ω(n)
- ( ) The worst case time complexity for both operations will be Ω(n)
- ( ) Worst case time complexity for both operations will be Ω(log n)

Question 22

0.2 / 0.2 pts

Suppose you are given a binary tree with n nodes, such that each node has exactly either zero or two children. The maximum height of the tree will be

- ( ) (n + 1) / 2

- n / 2 - 1
- n / 2 + 1
- (n – 1) / 2

Question 23

0.2 / 0.2 pts

Which of the following number of nodes can form a full binary tree?

- 8
- 14
- 15
- 13

Question 24

0.2 / 0.2 pts

A complete binary tree with the property that the value at each node is at least as large as the values at its children is known as

- completely balanced tree
- binary search tree
- (•) Heap
- AVL tree

Question 25

0.2 / 0.2 pts

Find the inorder and postorder of the binary tree with the given preorder: 60, 40, 20, 10, 30, 33, 50, 44, 51, 90, 70, 65, 80, 110, 100, 95, 99, 120.

- In order: 10, 33, 30, 20, 44, 51, 50, 40, 60, 65, 80, 70, 99, 95, 100, 120, 110, Postorder: 10, 20, 30, 33, 40, 44, 50, 51, 60, 65, 70, 80, 90, 95, 99, 100, 110
- In order: 10, 33, 30, 20, 44, 51, 60, 65, 80, 70, 99, 95, 100, 120, 110, Postorder: 110, 100, 99, 90, 80, 70, 65, 60, 51, 50, 44, 40, 33, 30, 20, 10.
- (•) Inorder: 10, 20, 30, 33, 40, 44, 50, 51, 60, 65, 70, 80, 90, 95, 99, 100, 110, 120 Postorder: 10, 33, 30, 20, 44, 51, 50, 40, 65, 80, 70, 99, 95, 100, 120, 110, 90, 60

In order: 110, 100, 99, 90, 80, 70, 65, 60, 51, 50, 44, 40, 33, 30, 20, 10. Postorder: 110, 120, 100, 95, 99, 70, 80, 65, 60, 40, 50, 51, 44, 20, 30, 33, 10

Quiz Score: **5** out of 5