IDMSQ08 – In an array-based stack, which operation has time complexity O(N) in the worst- case?
No operation that has time complexity O(N)

IDESQ07 in ADT of the Queue data structure, enqueue() method will

Add a new item to the queue at the rear position.

IDESQAS13 – Given the following input (4322, 1334, 1471, 9679, 1989, 6171, 6173, 4199) and the hash function: h(k)=k mod 10. Which of the following statements are true?
4199 and 9679 hash to the same value

IDMSQ03 – A stack S has 05 character items, S={"5", "4", "3", "2", "1"} where "1" is the top of S. Which operations must be perform to change S into a new state: S={"5", "4", "2", "3", "1"}?
pop()-->pop()-->pop()-->push("2")-->push("3")-->push("1")

IDESQAS14 – In the context of seach algorithms, which of the following statements are true? Choose one answer.
Binary search is faster than linear search, but it requires a sorted array.

Marks: 1

IDMAOA03 – What is the time complexity of the following algorithm with respect to the input size N

```
Algorithm: GCD(m,n)
Input: Two integers m and n
Output: The value of gcd(m,n)

i←n
While (i>1) do
  If (m is divisible by i) AND (n is divisible by i)
    Return i
  i--
Return 1
```

Choose one

O(N)

IDESOA09: Wrong heap data structure

In a min-heap the parent node value is always greater than or equal to its children's values.

IDETRE07 – What can you say about the following tree?
This is a binary tree.
IDEGRA07 – Which is the best describe of the graph below?
Unweighted, undirected, complete graph.

IDELI14 – Which is the common form of a node X in a Doubly Linked list

X(data, prev, next)

IDMAOA05 – What is the time complexity of the following algorithm with respect to the input size N

```
Algorithm sum(n)
Input: an integer n
Output: the sum S = ∑_{i=1}^{n} i^3

    s ← 0
    for  i← 1 to n do
         s=s+i*i*i;
    return s;
```

Choose one

O(N)

IDEGRA08 Which of the following is wrong about graph

Weigh of an edge must be possitive

IDESOA10 – In Merge sort algorithm …?

The input array is divided into two parts at the middle of the array

IDESOA01 - Which statement below is wrong in the context of sorting algorithms?
The sort key must be numeric

IDESQ16 – What is the result of the following operation on the stack S: S.peek(S.push(X))?

X.

IDMGRA01 –The maximum degree of any vertex in a simple graph with N vertices is?

N-1

1. Method search() is used to search for an item in a singly linked list. Please complete the code for this method?

```
public int search(int data)
{
   int count=1;
   SLNode current=this.head;
   while ((current !=null) && (current.getData() != [data] ))
   {
      count++;
      current=[current.getNext()] ;
   }
   if (current == null)
      return -1;

   else
```

```
        return count ;

  }
```

IDMSQ01 – A stack S has 05 character items, S={“A”,“B”,“C”,“D”,“E”} where “E” is the top of S. What is the content of S if we perform the following list of operations on the stack: push(“F”) –-> pop() –-> pop() –-> pop() –-> push(“D”)?
S={“A”,“B”,“C”,“D”}

IDESQAS04 – What additional requirement is placed on an array, so that binary search may be used to search for a key?
The array must be sorted

IDHAOA08 – The method f3(N) calls two methods f1(N) and f2(N) as follows. What is the time complexit of method f3(N)?

Question10

Marks: 1

IDHAOA08 – The method f3(N) calls two methods f1(N) and f2(N) as follows. What is

```
Suppose that f1(n) is O(n^2) and f2(n) is O(n)
public void f3(int n)
{
     for (int i=1; i<n; i++)
     {
          f2(n);
          for (int j=1; j<n; j++)
               f1(n);
     }
```

Choose one

O(N^4)

IDHSOA05 – Consider a modified version of Merge sort where the input array is partitioned at the position one-third of the length N of the array. What is the recurrence of this algorithm?

T(N)=T(N/3)+T(2N/3)+O(N)

IDESOA14 – Suppose that we are using Radix sort on N elements, each element has P digits in base b (each digit is in the range [0 .. B-1]), and couting sort algorithm is used to sort the digits. What is the time complexity of the Radix sort algorithm?
O(P(N+B))

IDESOA12 – In Radix sort algorithm …?

A stable sorting algorithm is used to sort the digits.

IDMTRE20 – In a binary search tree, node B is right child of node A, and node C is the right child of B. Which of the following statements are true?
Node C has the biggest value

IDEGRA09 – To implement Dijkstra's shortest path algorithm on unweighted graphs the data structure to be used is?
Queue

IDELI13– In a Circular Linked List, if a Node X(data,next) is a tail which is the value of the X's next?

Head
IDETRE04 – Number of leaf nodes in a perfect binary tree of depth h is?

2^h.

IDESQ11 – Which statement is wrong about list-based queue?

Queue is empty when front=rear.

IDESQAS07 – A close hashing hash table has an array size of 512. What is the maximum number of entries that can be placed in the table?
512

1. The following method reverses the item's order of a stack using a queue. Please complete the code of the method?

```
public static int reverse(SLLStack s)
{
    ArrayQueue q = new ArrayQueue();
    while (!s.isEmpty())
    {
        StackNode node = s.pop() ;
        q.enqueue(node.getData());
    }
    while ( !q.isEmpty() )
    {
        StackNode newnode = new StackNode(q.dequeue());
        s.push( new noc );
    }
}
```

IDESQAS10 – Consider a hash table of size seven, with starting index zero, and a hash function h(k)=(3k+ mod 7. What is the address of the key k=10?

6

1. The following method reverses the item's order of a stack using a queue. Please complete the code of the method?

```
public static int reverse(SLLStack s)
{
    ArrayQueue q = new ArrayQueue();
    while (!s.isEmpty())
```

```
        {
            StackNode node = [ s.pop() ] ;
            q.enqueue(node.getData());
        }
        while ( [ !q.isEmpty() ] )
        {
            StackNode newnode = new StackNode(q.dequeue());
            s.push( [ new noc ] );
        }
    }
```

IDESOA04 – The time complexity of an algorithm T(N) is estimated by couting the number of primitive operations.…?
The relative order of elements with equal keys are maintained.

IDESQAS11: complete the code

```
public int LinearSearch(int[] a, int key)
int i=0;
while (i<a.length)
{
  if (a[i]==key)
    return ____;
  i++;
}
return -1;
```

Answer: i

IDMGRA03 - In an unweighted, undirected connected graph, the shortest path from a node S to every other node is computed most efficiently, in terms of time complexity by?
Performing a BFS starting from S.

This method implement an O(N) algorithm to rearrange array x so that the left part is the elements that is smaller than p, the right part is the elements that is bigger than p. Please complete the code for this method?

```
public static void rearrange(int [] x, int p)
{
int left=0;
int righ[illegible]1;
while ( [ left<right ] )
{
[illegible][illegible]eft]<p)&&(left<x.length))
[ left++ ] ;
while ((x[right]>p)&&(right>=0))
```

right--
;
if (left<right)
{
int tmp=x[left];
x[left]=x[right];
x[right]=tmp;
}

IDEAOA05 – When evaluating algorithm's complexity, which approach makes possible an evaluation that independent of the hardware and software environments?

Theoretical approach

IDEAOA01 - Which statement below is correct?

Algorithm is a step-by-step procedure for solving a problem in a finite amount of time

IDELI07 – In a Singly Linked List, if a Node X(data,next) is a tail which is the value of the X's next?
null

IDESOA03 – Which statement below is wrong in the context of linear sorting algorithm?

The sort key must be numeric.

IDMLI04 – In a Singly Linked List implementation, what does this code to to the list?
Remove the node at the pos position from the list

IDMSOA12 – The Merge method in Merge sort algorithm is used to combine two sorted array A={3,27,38,43} and B={9,10,82}. What is the result array C?

C={3,9,10,27,38,43,82}

IDMAOA04 – What is the time complexity of the following algorithm with respect to the input size N

O(N)

IDELI04 – Which statement is correct about array-based list?

Array-based is faster than linked-list in case of accessing list's items.

IDELI03 – In the ADT of the list data structure, isEmpty() method returns a/an __ value?

Boolean.

IDMTRE06 – Given the following tree, what is the result of pre-order traversal the tree?

A,B,D,C,E,G,J,F,H,I

IDMSQ02 – A queue Q has 05 character items, Q={"A", "B", "C", "D", "E"} where "E" is the rear and "A" is the front of the queue. What is the content of Q if we perform the following list of operations on the queue: enqueue("F")-->dequeue()-->dequeue()--
>dequeque()--
>enqueue("D")?

Q={"D", "E,"F","D"}

IDMSOA03 – Bubble sort algorithm is used to sort the array A={23,78,45,8,32,56}
in the ascending order. What are the items of A after 03 sort pass?

A={8,23,32,45,56,78}

IDETRE17 – Which of the following is correct about array-based binary implementation

using perfect binary tree indexing scheme?
The left child and right child of node i are 2i+1 and 2i+2.

IDHTRE06 – Consider the recursive, nested representation of binary trees: T=(O L R) indicates a binary tree T with the root node O, the left sub-tree L and the right sub-tree R. Note that L and R may be null or further nested. Which of the following represents a valid binary tree?

(1 (2 3 4) (5 6 7))

IDMTRE07 – Given the following tree, what is the result of in-order traversal the tree?

B,D,A,G,J,E,C,H,F,I

IDETRE16 – Complete the following code of the method getNodeLabel() in the array-based tree implementation?
l[node]

IDESQ03 – In the ADT of the Stack data structure, push() method is used to?

add an item to the stack.

IDESOA08 – Which sorting algorithm scans and exchanges any pair of elements that is out-of-order

Bubble sort

IDESQAS08 – Which of the following is not an application of the queue data structure?

Evaluating a posfix expression.

IDELI11 – Complete the code below to insert a new node X at the POS position of a Singly Linked List?

Y.getNext().

1. The method below represent a number k in base b using a stack. Please complete the code of this method? (đúng ½ code -_-)

```
public void BaseConversion(int k, int b)
{
    ArrayStack s = new ArrayStack();
    while (k/b != 0)
    {
        k=k/b;
        ;
    }
    s.push(k);
    while (!s.isEmpty())
        System.out.print( s.pop() );
```