CT303-N: Data Structures & Algorithms

Assignment # 1 (Introduction)

1. _____ and _____ are the primary concerns for data structures and algorithms.
[A] Efficiency and Parallelism
[B] Parallelism and Scalability
[C] Efficiency and Scalability
[D] Preprocessing and Parallelism

2. Data structure is all about organizing, managing and storing data in such a way that it is efficiently handled.
[A] True
[B] False

3. Which of the following is NOT abstract data types. (i) List (ii) Queue (iii) Stack (iv) Tree
[A] (i), (ii) and (iii)
[B] (i) (iii) and (iv)
[C] All
[D] None

4. Which are the two parameters to measure computations complexity of any algorithm.
[A] Time and Bandwidth
[B] Time and Space
[C] Space and Bandwidth
[D] Number of iteration and input data

5. Which notation is used to measure the worst case computational complexity of an algorithm.
[A] Big Theta
[B] Big Omega
[C] Big O
[C] Big Alpha

6. How to measure the upper and lower bound of any algorithm?
[A] Using Best case and average case scenario
[B] Using Best case and worst case scenario
[C] Using normal case and worst case scenario
[D] Using normal case and average case scenario

7. Compare Primitive Data Structure with Non-Primitive Data Structure
8. Compare Static Data Structure with Dynamic Data Structure
9. Compare Linear Data Structure with Nonlinear Data Structure
10. List the common operations which can be performed on a typical data structure

CT303-N: Data Structures & Algorithms

Answer Key # 1 (Introduction)

1. [C], 2. [A], 3. [D], 4. [B], 5. [C], 6. [B]

7.

Primitive Data Structure	Non-Primitive Data Structure
Fundamental data structure available for programming	User defined data structure Classified into: linear and non-linear data structure
Example: int, float, char, pointer	Example of linear data structure: array, stack, queue, linked list Example of non-linear data structure: tree, graph
Can store only single data value	Can store multiple data values

8.

Static Data Structure	Dynamic Data Structure
Memory is allocated at compile time (before running/execution).	Memory is allocated at run time
Maximum size is fixed	Maximum size is flexible (can be increased/reduced)
Example: Array	Example: Linked list
Advantage: Fast access	Advantage: Faster insertion/deletion, Memory save
Disadvantage: Slower insertion/deletion, memory waste	Disadvantage: Slow access, Computation overhead

9.

Linear Data Structure	Nonlinear Data Structure
It arranges the data in orderly manner where the elements are attached adjacently	It arranges the data in sorted order, creating a relationship among the data element
Memory Utilization: Inefficient	Memory Utilization: Efficient
Level: Single	Level: Multiple
Implementation: Easier	Implementation: Difficult
Example: Array, Stack, Linked List, Queue	Example: Tree, Graph

10. Search/Traverse, Sort, Insert, Update, Delete

CT303-N: Data Structures & Algorithms

Assignment # 2 (Sparse Matrix & Stack)

1. Which matrix has most of the elements (not all) as Zero?
[A] Identity Matrix
[B] Unit Matrix
[C] Sparse Matrix
[D] Zero Matrix

2. The matrix contains m rows and n columns. The matrix is called Sparse Matrix if _____
[A] Total number of Zero elements > (m*n)/2
[B] Total number of Zero elements = m + n
[C] Total number of Zero elements = m/n
[D] Total number of Zero elements = m-n

3. What is the difference between a normal(naive) array and a sparse array?
[A] Sparse array can hold more elements than a normal array
[B] Sparse array is memory efficient
[C] Sparse array is dynamic
[D] A naive array is more efficient

4. Suppose the contents of an array A are, A = {1, null, null, null, null, 10};
What would be the size of the array considering it as a normal array and a sparse array?
[A] 6 and 6
[B] 6 and 2
[C] 2 and 6
[D] 2 and 2

5. What is sparsity of a matrix?
[A] The fraction of zero elements over the total number of elements
[B] The fraction of non-zero elements over the total number of elements
[C] The fraction of total number of elements over the zero elements
[D] The fraction of total number of elements over the non-zero elements

6. Which of the following is the disadvantage of sparse matrices over normal matrices?
[A] Size
[B] Speed
[C] Easily compressible
[D] Algorithm complexity

7. Stack works on the principle of
[A] First in first out
[B] Last in first out
[C] Both A and B
[D] None of above

8. Process of inserting an element in stack is called _____
[A] Create
[B] Push
[C] Evaluation
[D] Pop

9. Process of removing an element from stack is called _____
[A] Create
[B] Push

[C] Evaluation
[D] Pop

10. In a stack, if a user tries to remove an element from empty stack it is called _____
[A] Underflow
[B] Empty collection
[C] Overflow
[D] Garbage Collection

11. Pushing an element into stack already having five elements and stack size of 5, then stack becomes
[A] Overflow
[B] Crash
[C] Underflow
[D] User flow

12. Entries in a stack are "ordered". What is the meaning of this statement?
[A] A collection of stacks is sortable
[B] Stack entries may be compared with the '<' operation
[C] The entries are stored in a linked list
[D] There is a Sequential entry that is one by one

13. Consider the following operation performed on a stack of size 5.
Push(1);
Pop();
Push(2);
Push(3);
Pop();
Push(4);
Pop();
Pop();
Push(5);

After the completion of all operation, the number of element present on stack are
[A] 1
[B] 2
[C] 3
[D] 4

14. If the elements "A", "B", "C" and "D" are placed in a stack and are deleted one at a time, in what order will they be removed?

[A] ABCD
[B] DCBA
[C] DCAB
[D] ABDC

15. A single array A[1..MAXSIZE] is used to implement two stacks. The two stacks grow from opposite ends of the array. Variables top1 and top2 (topl< top 2) point to the location of the topmost element in each of the stacks. If the space is to be used efficiently, the condition for "stack full" is (GATE CS 2004)
[A] (top1 = MAXSIZE/2) and (top2 = MAXSIZE/2+1)
[B] top1 + top2 = MAXSIZE
[C] (top1= MAXSIZE/2) or (top2 = MAXSIZE)

[D] top1= top2 -1

16. The seven elements A, B, C, D, E, F and G are pushed onto a stack in reverse order, i.e., starting from G. The stack is popped five times and each element is inserted into a queue.Two elements are deleted from the queue and pushed back onto the stack. Now, one element is popped from the stack. The popped item is _____.
[A] A
[B] B
[C] F
[D] G

17. The five items: A, B, C, D, and E are pushed in a stack, one after other 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
[A] A
[B] B
[C] C
[D] D

18. Consider the following operations performed on a stack of size 5 : Push (a); Pop() ; Push(b); Push(c); Pop(); Push(d); Pop();Pop(); Push (e) Which of the following statements is correct?
[A] Underflow occurs
[B] Stack operations are performed smoothly
[C] Overflow occurs
[D] None of the above

19. The five items: A, B, C, D, and E are pushed in a stack, one after other 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
[A] A
[B] B
[C] C
[D] D

Answer Key # 2 (Sparse Matrix & Stack)

1. [C], 2. [A], 3. [B], 4. [B], 5. [A], 6. [D], 7. [B], 8. [B], 9. [D], 10. [A], 11. [A], 12. [D], 13. [A], 14. [B]
15. [D], 16. [B], 17. [D], 18. [B], 19. [D]

CT303-N: Data Structures & Algorithms

Laboratory Exercise # 1

(August 3, 2020)

1.1: Write a program to check whether the given matrix is sparse matrix or not.
(Hint: https://github.com/hbpatel1976/Data-Structure/blob/master/sparse1.c)

1.2: Write a program to convert the given matrix into sparse matrix.
(Hint: https://github.com/hbpatel1976/Data-Structure/blob/master/sparse2.c)

1.3: Write a program to experiment PUSH and POP operations on a stack.
(Hint: https://github.com/hbpatel1976/Data-Structure/blob/master/stack.c)

1.4: Write a complete menu driven program that incorporates all the functionalities of the stack viz. push, pop, peep, isFull, isEmpty.
(Hint: https://github.com/hbpatel1976/Data-Structure/blob/master/stackfull.c)

CT303-N: Data Structures & Algorithms

Assignment # 3 (Reverse Polish Notation)

1. What is the other name for a postfix expression?
[A] Normal polish Notation
[B] Reverse polish Notation
[C] Warsaw notation
[D] Infix notation

2. Reverse Polish Notation is the reverse of a Polish Notation
[A] True
[B] False

3. Which of the following is not an application of stack?
[A] evaluation of postfix expression
[B] conversion of infix to postfix expression
[C] line at ticket counter

4. When an operand is read, which of the following is done?
[A] It is placed on to the output
[B] It is placed in operator stack
[C] It is ignored
[D] Operator stack is emptied

5. What should be done when a left parenthesis '(' is encountered?
[A] It is ignored
[B] It is placed in the output
[C] It is placed in the operator stack
[D] The contents of the operator stack is emptied

6. Which of the following is an infix expression?
[A] (a+b)*(c+d)
[B] ab+c*
[C] +ab
[D] abc+*

7. What is the postfix expression for the corresponding infix expression a+b*c+(d*e)?
[A] abc*+de*+
[B] abc+*de*+
[C] a+bc*de+*
[D] abc*+(de)*+

8. Parentheses are simply ignored in the conversion of infix to postfix expression.
[A] True
[B] False

9. It is easier for a computer to process a postfix expression than an infix expression.
[A] True
[B] False

10. What is the postfix expression for the infix expression a-b-c?
[A] -ab-c
[B] ab – c –
[C] – -abc
[D] -ab-c

11. What is the postfix expression for the following infix expression a/b^c-d?
[A] abc^/d-
[B] ab/cd^-
[C] ab/^cd-
[D] abcd^/-

12. Which of the following statement is incorrect with respect to infix to postfix conversion algorithm?
[A] operand is always placed in the output
[B] operator is placed in the stack when the stack operator has lower precedence
[C] parenthesis are included in the output
[D] higher and equal priority operators follow the same condition

13. In infix to postfix conversion algorithm, the operators are associated from?
[A] right to left
[B] left to right
[C] centre to left
[D] centre to right

14. What is the corresponding postfix expression for the given infix expression a*(b+c)/d?
[A] ab*+cd/
[B] ab+*cd/
[C] abc*+/d
[D] abc+*d/

15. What is the result of the given postfix expression? abc*+ where a=1, b=2, c=3.
[A] 4
[B] 5
[C] 6
[D] 7

16. What is the result of the following postfix expression?
ab*cd*+ where a=2,b=2,c=3,d=4.
[A] 16
[B] 12
[C] 14
[D] 10

17. Consider the stack
| 5 |
| 4 |
| 3 |
| 2 |.
At this point, '*' is encountered. What has to be done?
[A] 5*4=20 is pushed into the stack
[B] * is pushed into the stack
[C] 2*3=6 is pushed into the stack
[D] * is ignored

18. Evaluate the postfix expression ab + cd/- where a=5, b=4, c=9, d=3.
[A] 23
[B] 15
[C] 6
[D] 10

CT303-N: Data Structures & Algorithms

Answer Key Assignment # 3 (Reverse Polish Notation)

1 [B], 2 [B], 3 [C], 4 [A], 5 [C], 6 [A], 7 [A], 8 [B], 9 [A], 10 [B], 11 [A], 12 [C], 13 [B], 14 [D], 15 [D], 16 [A], 17 [A], 18 [C]

CT303-N: Data Structures & Algorithms

Assignment # 4 (Recursion)

1. Recursion is a method in which the solution of a problem depends on _____
[A] Larger instances of different problems
[B] Larger instances of the same problem
[C] Smaller instances of the same problem
[D] Smaller instances of different problems

2. Which of the following problems can't be solved using recursion?
[A] Factorial of a number
[B] Fibonacci number
[C] Exponent
[D] Problems without base case or terminating condition

3. Recursion is similar to which of the following?
[A] Switch Case
[B] Loop
[C] If-else
[D] if else if else

4. In recursion, the condition for which the function will stop calling itself is _____
[A] Best case
[B] Worst case
[C] Base case
[D] There is no such condition

5. Consider the following code snippet:

```
void my_recursive_function()
{
  my_recursive_function();
}
int main()
{
  my_recursive_function();
  return 0;
}
```

What will happen when the above snippet is executed?
[A] The code will be executed successfully and no output will be generated
[B] The code will be executed successfully and random output will be generated
[C] The code will show a compile time error
[D] The code will run for some time and stop when the stack overflows

6. What is the output of the following code?

```
void my_recursive_function(int n)
{
   if(n == 0)
   return;
   printf("%d ",n);
   my_recursive_function(n-1);
}
int main()
{
   my_recursive_function(10);
   return 0;
```

```
}
```
[A] 10
[B] 1
[C] 10 9 8 … 1 0
[D] 10 9 8 … 1

7. What is the base case for the following code?
```
void my_recursive_function(int n)
{
    if(n == 0)
    return;
    printf("%d ",n);
    my_recursive_function(n-1);
}
int main()
{
    my_recursive_function(10);
    return 0;
}
```
[A] return
[B] printf(“%d “, n)
[C] if(n == 0)
[D] my_recursive_function(n-1)

8. What will be the output of the following code?
```
int cnt=0;
void my_recursive_function(int n)
{
    if(n == 0)
    return;
    cnt++;
    my_recursive_function(n/10);
}
int main()
{
    my_recursive_function(123456789);
    printf("%d",cnt);
    return 0;
}
```
[A] 123456789
[B] 10
[C] 0
[D] 9

9. What will be the output of the following code?
```
void my_recursive_function(int n)
{
   if(n == 0)
   {
       printf("False");
            return;
   }
```

```
   if(n == 1)
   {
       printf("True");
       return;
   }
   if(n%2==0)
   my_recursive_function(n/2);
   else
   {
       printf("False");
       return;
   }
}
int main()
{
    my_recursive_function(100);
    return 0;
}
```

[A] True
[B] False

10. What is the output of the following code?

```
int cnt = 0;
void my_recursive_function(char *s, int i)
{
    if(s[i] == '\0')
      return;
    if(s[i]=='a' || s[i]=='e' || s[i]=='i' || s[i]=='o' || s[i]=='u')
    cnt++;
    my_recursive_function(s,i+1);
}
int main()
{
    my_recursive_function("thisisrecursion",0);
    printf("%d",cnt);
    return 0;
}
```

[A] 6
[B] 9
[C] 5
[D] 10

11. What will be the output of the following C code?

```
#include<stdio.h>
main()
{
   int n;
   n=f1(4);
   printf("%d",n);
}
f1(int x)
{
   int b;
```

```
   if(x==1)
      return 1;
   else
      b=x*f1(x-1);
      return b;
}
```

[A] 24
[B] 4
[C] 12
[D] 10

12. The data structure used to implement recursive function calls _____
[A] Array
[B] Linked list
[C] Binary tree
[D] Stack

13. What will be the output of the following C code?

```
#include<stdio.h>
main()
{
   int n,i;
   n=f(6);
   printf("%d",n);
}
f(int x)
{
   if(x==2)
         return 2;
   else
   {
      printf("+");
      f(x-1);
   }
}
```

[A] ++++2
[B] +++++2
[C] +++++
[D] 2

14. How many times is 'a' printed when the following C code is executed?

```
#include<stdio.h>
main()
{
   int a;
   a=f1(10);
   printf("%d",a);
}
f1(int b)
{
   if(b==0)
      return 0;
   else
```

```
   {
      printf("a");
      f1(b--);
   }
}
```

[A] 9 times
[B] 10 times
[C] 0 times
[D] Infinite number of times

15. What will be the output of the following C code?

```
#include<stdio.h>
main()
{
   int n=10;
   int f(int n);
   printf("%d",f(n));
}
int f(int n)
{
   if(n>0)
      return(n+f(n-2));
}
```

[A] 10
[B] 80
[C] 30
[D] Error

16.What will be the output of the following C code if the input given to the code shown below is “VSITR”?

```
#include<stdio.h>
#define NL '\n'
main()
{
   void f(void);
   printf("enter the word\n");
   f();
}
void f(void)
{
   char c;
   if((c=getchar())!=NL)
{
f();
   printf("%c",c);
}
return;
}
```

[A] VSITR
[B] infinite loop
[C] RTISV
[D] RITVS

17. What does the following function print for n = 25?

```
void fun(int n)
{
  if (n == 0)
    return;

  printf("%d", n%2);
  fun(n/2);
}
```

[A] 11001
[B] 10011
[C] 11111
[D] 00000

18. What does the following function do?

```
int fun(int x, int y)
{
   if (y == 0)   return 0;
   return (x + fun(x, y-1));
}
```

[A] x + y
[B] x + x*y
[C] x*y
[D] x^y

19. What happens if base condition is not defined in recursion?
[A] Stack underflow
[B] Stack Overflow
[C] None of these
[D] Both a and b

Answer Key Assignment # 4 (Recursion)

1 [C], 2 [D], 3 [B], 4 [C], 5 [D], 6 [D], 7 [C], 8 [D], 9 [B], 10 [A], 11 [A], 12 [D], 13 [A], 14 [D], 15 [C], 16 [C], 17 [B], 18 [C], 19 [B]

CT303-N: Data Structures & Algorithms

Laboratory Exercise # 2

(August 17 & 18, 2020)

2.1 Write a program to convert given infix expression into postfix expression.
(https://github.com/hbpatel1976/Data-Structure/blob/master/in2post.c)

2.2 Write a recursive program to compute

- 2.2.1 Factorial (https://github.com/hbpatel1976/C-Programming/blob/master/Ch0608.c)
- 2.2.2 Exponent ^{base} (https://github.com/hbpatel1976/C-Programming/blob/master/Ch0609.c)
- 2.2.3 Fibonacci series (https://github.com/hbpatel1976/C-Programming/blob/master/Ch0610.c)

2.3 Implement Queue operations (Insert, Delete, Display etc.) using arrays.
https://github.com/hbpatel1976/Data-Structure/blob/master/queue1.c

Data Structures and Algorithms

Laboratory Exercise # 3

(August 24 & 25, 2020)

3.1 Implement circular queue using arrays.
(*https://github.com/hbpatel1976/Data-Structure/blob/master/cqueue.c*)

3.2 Implement double ended queue using arrays.
(https://github.com/hbpatel1976/Data-Structure/blob/master/dequeue.c)

3.3 Implement priority queue using arrays.
(https://github.com/hbpatel1976/Data-Structure/blob/master/pqueue.c)

CT303-N: Data Structures & Algorithms

Assignment # 4 (Queue)

1: A linear list of elements in which deletion can be done from one end (front) and insertion can take place only at the other end (rear) is known as a?
[A] Queue
[B] Stack
[C] Tree
[D] Linked list

2: A queue follows _____
[A] FIFO (First In First Out) principle
[B] LIFO (Last In First Out) principle
[C] Ordered array
[D] Linear tree

3: If the elements “A”, “B”, “C” and “D” are placed in a queue and are deleted one at a time, in what order will they be removed?
[A] DCBA
[B] DCAB
[C] ABDC
[D] ABCD

4: A normal queue, if implemented using an array of size MAX_SIZE, gets full when
[A] Front = (rear + 1)mod MAX_SIZE
[B] Rear = MAX_SIZE – 1
[C] Front = rear + 1
[D] Rear = front

5. Queues serve major role in _____
[A] Simulation of recursion
[B] Simulation of arbitrary linked list
[C] Simulation of limited resource allocation
[D] Simulation of heap sort

6: Which of the following is not the type of queue?
[A] Ordinary/Regular queue
[B] Single ended queue
[C] Circular queue
[D] Priority queue

7: The value of REAR is increased by 1 when …….
[A] An element is deleted in a queue
[B] An element is traversed in a queue
[C] An element is added in a queue
[D] An element is merged in a queue

8: An array of size MAX_SIZE is used to implement a circular queue. Front, Rear, and count are tracked. Suppose front is 0 and rear is MAX_SIZE -1. How many elements are present in the queue?
[A] Zero
[B] One
[C] MAX_SIZE-1
[D] MAX_SIZE

9: Circular Queue is also known as _____
[A] Ring Buffer
[B] Square Buffer
[C] Rectangle Buffer
[D] Curve Buffer

10: The operations that can be done in a circular queue is/are …..
[A] Insert from the front end
[B] Delete from front end
[C] Display queue contents
[D] All of the above

11: If the MAX_SIZE is the size of the array used in the implementation of circular queue. How is rear manipulated while inserting an element in the queue?
[A] rear=(rear%1)+MAX_SIZE
[B] rear=rear%(MAX_SIZE+1)
[C] rear=(rear+1)%MAX_SIZE
[D] rear=rear+(1%MAX_SIZE)

12: If the MAX_SIZE is the size of the array used in the implementation of circular queue, array index start with 0, front point to the first element in the queue, and rear point to the last element in the queue. Which of the following condition specify that circular queue is FULL?
[A] Front=rear= -1
[B] Front=(rear+1)%MAX_SIZE
[C] Rear=front+1
[D] Rear=(front+1)%MAX_SIZE

13: The condition ____ indicate the queue is empty.
[A] Front=Null
[B] Front=SIZE
[C] Front=Rear
[D] Rear=Null

14: What is a dequeue?
[A] A queue with insert/delete defined for both front and rear ends of the queue
[B] A queue implemented with a doubly linked list
[C] A queue implemented with both singly and doubly linked lists
[C] A queue with insert/delete defined for front side of the queue

15: The another name of dequeue is ……
[A] divided queue
[B] distributed queue
[C] double ended queue
[D] design queue

16: Identify the data structure which allows deletions at both ends of the list but insertion at only one end.
[A] Input restricted dequeue
[B] Output restricted qequeue
[C] Priority queues
[D] Stack

17: _____ is a collection of elements such that each element has been assigned a processing priority.
[A] Priority queue

[B] Procedure queue
[C] Main queue
[D] Interrupt queue

18: After performing these set of operations, what does the final list look contain?
InsertFront(10);
InsertFront(20);
InsertRear(30);
DeleteFront();
InsertRear(40);
InsertRear(10);
DeleteRear();
InsertRear(15);
display();

[A] 10 30 10 15
[B] 20 30 40 15
[C] 20 30 40 10
[D] 10 30 40 15

19: In a priority queue, insertion and deletion takes place at ……
[A] front, rear end
[B] only at rear end
[C] only at front end
[D] any position

Answer Key Assignment # 4 (Queue)

1: [A], 2: [A], 3: [D], 4: [B], 5: [C], 6: [B], 7: [C], 8: [D], 9: [A], 10: [D], 11: [C], 12: [B], 13: [A], 14: [A], 15: [C]
16: [A], 17: [A], 18: [D], 19: [D]

Midterm Questions:

1. Differentiate Linear Vs Non-Linear Data structure with appropriate figures.
2. Let S is an instance of stack. Consider the following sequence of operations performed on S which initially contains element with 55 as top most elements. What will be the element at the top of the stack after the following sequence of operations?
S.push(100), S.pop(), S.push(200), S.push(300), S.pop() [Explain with appropriate diagram]
3. Convert the infix expression (x^y*z)+(a-b/c) into postfix notation. [Explain the steps]
4. Write a function enqueue(x) to insert an element an a circular queue. [Use any programming language]
5. Write functions push(x) and pop to insert and delete elements from stack, respectively. [Use any programming language]

CT303-N: Data Structures & Algorithms

Assignment # 5 (Linked List)

1. A linear collection of data elements where the linear node is given by means of pointer is called?
[A] Linked list
[B] Node list
[C] Primitive list
[D] None

2. In linked list each node contains minimum of two fields. One field is data field to store the data second field is?
[A] Pointer to character
[B] Pointer to integer
[C] Pointer to node
[D] Node

3. What does creating a node mean?
[A] Defining its structure
[B] Allocating memory to it
[C] Initialization
[D] All of the above

4. Which of these is not an application of linked list?
[A] To implement file systems
[B] For separate chaining in hash-tables
[C] To implement non-binary trees
[D] Random Access of elements

5. Which of the following statements about linked list data structure is/are TRUE?
[A] Addition and deletion of an item to/ from the linked list require modification of the existing pointers
[B] The linked list pointers do not provide an efficient way to search an item in the linked list
[C] Linked list pointers always maintain the list in ascending order
[D] The linked list data structure provides an efficient way to find kth element in the list

6. The situation when in a linked list START=NULL is
[A] underflow
[B] overflow
[C] housefull
[D] saturated

7. Consider the following statements:
(I) First-in-first out types of computations are efficiently supported by STACKS.
(II) Implementing LISTS on linked lists is more efficient than implementing LISTS on an array for almost all the basic LIST operations.
(III) Implementing QUEUES on a circular array is more efficient than implementing QUEUES on a linear array with two indices.
(IV) Last-in-first-out type of computations are efficiently supported by QUEUES.
Which of the following is correct?
[A] (II) and (III) are true
[B] (I) and (II) are true
[C] (III) and (IV) are true
[D] (II) and (IV) are true

8. Consider the following definition in c programming language
struct node

```
{
int data;
struct node * next;
}
struct node *ptr;
```

Which of the following c code is used to create new node?
[A] ptr=(node*)malloc(sizeof(node));
[B] ptr=(node*)malloc(node);
[C] ptr=(node*)malloc(sizeof(node*));
[D] ptr=(node)malloc(sizeof(node));

9. Which of the following points is/are true about Linked List data structure when it is compared with array?
[A] Arrays have better cache locality that can make them better in terms of performance.
[B] It is easy to insert and delete elements in Linked List
[C] Random access is not allowed in a typical implementation of Linked Lists
[D] The size of array has to be pre-decided, linked lists can change their size any time.
[E] All of the above

10. You are given pointers to first and last nodes of a singly linked list, which of the following operations are dependent on the length of the linked list?
[A] Delete the first element
[B] Insert a new element as a first element
[C] Delete the last element of the list
[D] Add a new element at the end of the list

11. Consider an implementation of unsorted singly linked list. Suppose it has its representation with a head and tail pointer. Given the representation, which of the following operation can be implemented in O(1) time?
I) Insertion at the front of the linked list
II) Insertion at the end of the linked list
III) Deletion of the front node of the linked list
IV) Deletion of the last node of the linked list
[A] I and II
[B] I and III
[C] I,II and III
[D] I,II and IV

12. Consider an implementation of unsorted singly linked list. Suppose it has its representation with a head pointer only. Given the representation, which of the following operation can be implemented in O(1) time?
I) Insertion at the front of the linked list
II) Insertion at the end of the linked list
III) Deletion of the front node of the linked list
IV) Deletion of the last node of the linked list
[A] I and II
[B] I and III
[C] I,II and III
[D] I,II and IV

13. What is the output of following function for start pointing to first node of following linked list? 1->2->3->4->5->6

```
void fun(struct node* start)
{
  if(start == NULL)
    return;
```

```
  printf("%d  ", start->data);

  if(start->next != NULL )
    fun(start->next->next);
  printf("%d  ", start->data);
}
```

[A] 1 4 6 6 4 1
[B] 1 3 5 1 3 5
[C] 1 2 3 5
[D] 1 3 5 5 3 1

14. In the worst case, the number of comparisons needed to search a singly linked list of length n for a given element is ____
[A] log 2 n
[B] n/2
[C] log 2 n – 1
[D] n

15. Consider an implementation of unsorted single linked list. Suppose it has its representation with a head and a tail pointer (i.e. pointers to the first and last nodes of the linked list). Given the representation, which of the following operation cannot be implemented in O(1) time ?
[A] Insertion at the front of the linked list.
[B] Insertion at the end of the linked list.
[C] Deletion of the front node of the linked list.
[D] Deletion of the last node of the linked list.

16. A variant of linked list in which last node of the list points to the first node of the list is?
[A] Singly linked list
[B] Doubly linked list
[C] Circular linked list
[D] Multiply linked list

17. A variant of the linked list in which none of the node contains NULL pointer is?
[A] Singly Linked List
[B] Doubly Linked List
[C] Circular Linked List
[D] None

18. A circularly linked list is used to represent a Queue. A single variable p is used to access the Queue. To which node should p point such that both the operations enQueue and deQueue can be performed in constant time?
[A] rear node
[B] front node
[C] not possible with a single pointer
[D] node next to front

19. In circular linked list, insertion of node requires modification of?
[A] One pointer
[B] Two pointer
[C] Three pointer
[D] None

20. In a circular linked list
[A] Components are all linked together in some sequential manner.

[B] There is no beginning and no end.
[C] Components are arranged hierarchically.
[D] Forward and backward traversal within the list is permitted.

21. Consider an implementation of unsorted circular linked list. Suppose it has its representation with a head pointer only. Given the representation, which of the following operation can be implemented in O(1) time?
I) Insertion at the front of the linked list
II) Insertion at the end of the linked list
III) Deletion of the front node of the linked list
IV) Deletion of the end node of the linked list
[A] I and II
[B] I and III
[C] I, II, III and IV
[D] None

22. Which of the following operations is performed more efficiently by doubly linked list than by singly linked list?
[A] Deleting a node whose location in given
[B] Searching of an unsorted list for a given item
[C] Inverting a node after the node with given location
[D] Traversing a list to process each node

23. In doubly linked lists, traversal can be performed?
[A] Only in forward direction
[B] Only in reverse direction
[C] In both directions
[D] None

24. Consider an implementation of unsorted doubly linked list. Suppose it has its representation with a head pointer and tail pointer. Given the representation, which of the following operation can be implemented in O(1) time?
I) Insertion at the front of the linked list
II) Insertion at the end of the linked list
III) Deletion of the front node of the linked list
IV) Deletion of the end node of the linked list
[A] I and II
[B] I and III
[C] I,II and III
[D] I,II,III and IV

25. Consider an implementation of unsorted doubly linked list. Suppose it has its representation with a head pointer only. Given the representation, which of the following operation can be implemented in O(1) time?
I) Insertion at the front of the linked list
II) Insertion at the end of the linked list
III) Deletion of the front node of the linked list
IV) Deletion of the end node of the linked list
[A] I and II
[B] I and III
[C] I,II and III
[D] I,II,III and IV

26. In a doubly linked list, the number of pointers affected for an insertion operation will be
[A] 4
[B] 0

[C] 1
[D] None of these

27. Consider an implementation of unsorted circular doubly linked list. Suppose it has its representation with a head pointer only. Given the representation, which of the following operation can be implemented in O(1) time?
I) Insertion at the front of the linked list
II) insertion at the end of the linked list
III) Deletion of the front node of the linked list
IV) Deletion of the end node of the linked list
[A] I and II
[B] I and III
[C] I, II and III
[D] I,II,III and IV

28. What does the following function do for a given Linked List with first node as "start"?

```
void fun1(struct node* start)
{
  if(start == NULL)return;
  fun1(start->next);
  printf("%d  ", start->data);
}
```

[A] Prints all nodes of linked lists i normal order
[b] Prints all nodes of linked list in reverse order
[C] Prints alternate nodes of Linked List
[D] Prints alternate nodes in reverse order

Answer Key: Assignment # 5 (Linked List)

1: [A], 2. [C], 3. [B], 4. [D], 5. [B], 6. [A], 7. [A], 8. [A], 9. [E], 10. [C], 11. [C], 12. [B], 13. [D], 14. [D], 15. [D]
16. [C], 17. [C], 18. [A], 19. [B], 20. [B], 21. [D], 22. [A], 23. [C], 24. [D], 25. [B], 26. [D], 27. [D], 28. [A]

CT303-N: Data Structures & Algorithms

Laboratory Exercise # 4

(September 21 & 22, 2020)

4.1 Implement singly linked list.
(*https://github.com/hbpatel1976/Data-Structure/blob/master/llist1.c*)

4.2 Implement insertion (from beginning, end and intermediate) into a singly linked list.
(https://github.com/hbpatel1976/Data-Structure/blob/master/llinsert.c)

4.3 Implement deletion (from beginning, end and specific position) from a singly linked list.
(https://github.com/hbpatel1976/Data-Structure/blob/master/lldelete.c)

4.4 Implement circular (singly) linked list.
(https://github.com/hbpatel1976/Data-Structure/blob/master/llcircular.c)

4.5 Implement doubly linked list.
(https://github.com/hbpatel1976/Data-Structure/blob/master/lldoubly.c)

4.6 Implement doubly circular linked list.
(https://github.com/hbpatel1976/Data-Structure/blob/master/lldblcirc.c)

4.7 Implement STACK using singly linked list.
(https://github.com/hbpatel1976/Data-Structure/blob/master/stackll.c)

4.8 Implement QUEUE using singly linked list.
(https://github.com/hbpatel1976/Data-Structure/blob/master/queuell.c)

CT303-N: Data Structures & Algorithms

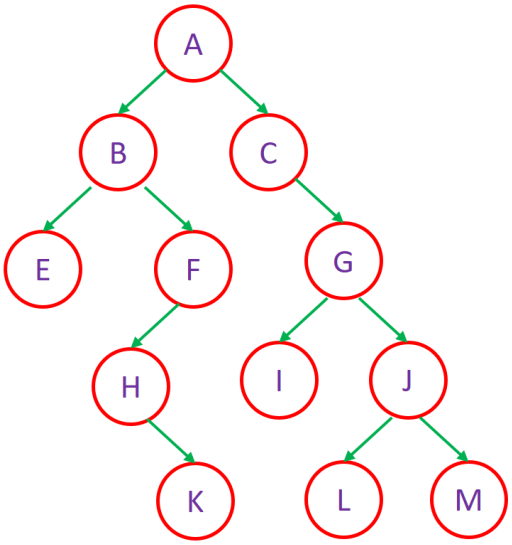
Laboratory Exercise # 5

(October 5 & 6, 2020)

5.1 To create a binary tree.
(*https://github.com/hbpatel1976/Data-Structure/blob/master/bintree.c*)

5.2 Implement all the three tree traversals (Preorder, Inorder and Postorder).
(https://github.com/hbpatel1976/Data-Structure/blob/master/btreetraverse.c)

5.3 Convert following binary tree into threaded binary tree.



5.4 Create binary search tree (BST) for following sequence of numbers:
12, 7, 9, 10, 18, 14, 1, 45, 89, 100, 36 59

5.5 In the binary search tree (BST) created in 5.4, delete (a) node with no child (b) node with one child and (c) node with two children.

5.6 Construct a BST from Preorder Data (50 25 12 30 27 75 10 90 80 83 125 112)

CT303-N: Data Structures & Algorithms

Assignment # 6 (Tree - I)

Binary Tree

1. The number of edges from the root to the node is called _____ of the tree.
(a) Height
(b) Depth
(c) Length
(d) Width

2. The number of edges from the node to the deepest leaf is called _____ of the tree.
(a) Height
(b) Depth
(c) Length
(d) Width

3. Which type of traversal of binary search tree outputs the value in sorted order?
(a) Pre-order
(b) In-order
(c) Post-order
(d) None

4. Select the one FALSE statement about binary trees
(a) Every binary tree has at least one node
(b) Every non-empty tree has exactly one root node
(c) Every node has at most two children
(d) Every non-root node has exactly one parent.

5. A Binary Tree can have
(a) Can have 2 children
(b) Can have 1 children
(c) Can have 0 children
(d) All

6. A binary tree T has n leaf nodes. The number of nodes of degree 2 in T is
(a) log2n
(b) n-1
(c) n
(d) 2n

7. Which of the following is not an advantage of trees?
(a) Hierarchical structure
(b) Faster search
(c) Router algorithms
(d) Undo/Redo operations in a notepad

8. What are the main applications of tree data structure? 1) Manipulate hierarchical data 2) Make information easy to search (see tree traversal). 3) Manipulate sorted lists of data 4) Router algorithms 5) Form of a multi-stage decision-making, like Chess Game. 6) As a workflow for compositing digital images for visual effects
(a) 1, 2, 3, 4 and 6
(b) 1, 2, 3, 4 and 5
(c) 1, 3, 4, 5 and 6
(d) 1, 2, 3, 4, 5 and 6

CT303-N: Data Structures & Algorithms

Binary Search Tree

9. A binary search tree is generated by inserting in order the following integers: 50, 15, 62, 5, 20, 58, 91, 3, 8, 37, 60, 24 The number of the node in the left sub-tree and right sub-tree of the root, respectively, is
(a) (4, 7)
(b) (7, 4)
(c) (8, 3)
(d) (3, 8)

Threaded Binary Tree

10. What are the disadvantages of normal binary tree traversals?
(a) there are many pointers which are null and thus useless
(b) there is no traversal which is efficient
(c) improper traversals
(d) complexity in implementing

11. What may be the content of a node in threaded binary tree?
(a) leftchild_pointer, left_tag, data
(b) leftchild_pointer, left_tag, data, right_tag, rightchild_pointer
(c) leftchild_pointer, left_tag
(d) leftchild_pointer, left_tag, right_tag, rightchild_pointer

12. For the inorder traversal of threaded binary tree, we introduced a dummy node. The left pointer of the dummy node is pointing to the _____ node of the tree.
(a) left most
(b) root
(c) right most
(d) any of the given node

13. In threaded binary tree the NULL pointers are replaced by the
(a) preorder successor or predecessor
(b) inorder successor or predecessor
(c) inorder successor or predecessor
(d) NULL pointers are not replaced

14. What are null nodes filled with in a threaded binary tree?
(a) inorder predecessor for left node and inorder successor for right node information
(b) right node with inorder predecessor and left node with inorder successor information
(c) they remain null
(d) some other values randomly

15. A Threaded Binary Tree is a binary tree in which every node that does not have a right child has a THREAD (in actual sense, a link) to its _____ successor
(a) Preorder
(b) Inorder
(c) Postorder
(d) levelorder

16. In delete operation of BST, we need inorder successor (or predecessor) of a node when the node to be deleted has both left and right child as non-empty. Which of the following is true about inorder successor needed in delete operation?
(a) Inorder Successor is always a leaf node
(b) Inorder successor is always either a leaf node or a node with empty left child

(c) Inorder successor may be an ancestor of the node
(d) Inorder successor is always either a leaf node or a node with empty right child Binary Search Trees

AVL Tree

17. Every AVL is _____
(a) Binary Tree
(b) Complete Binary Tree
(c) None of these
(d) Binary Search Tree

18. What is an AVL tree?
(a) a tree which is balanced and is a height balanced tree
(b) a tree which is unbalanced and is a height balanced tree
(c) a tree with three children
(d) a tree with atmost 3 children

19. True statements about AVL tree are
(a) It is a binary search tree
(b) Left node and right node differs in height by at most 1 unit
(c) Worst case time complexity is O(log2n)
(d) All above

20. Given an empty AVL tree, how would you construct AVL tree when a set of numbers are given without performing any rotations?
(a) just build the tree with the given input
(b) find the median of the set of elements given, make it as root and construct the tree
(c) use trial and error
(d) use dynamic programming to build the tree

21. What is the maximum height of any AVL-tree with 7 nodes? Assume that the height of a tree with a single node is 0.
(a) 2
(b) 3
(c) 4
(d) 5

Data Structures and Algorithms
Answer Key - Assignment # 6 (Tree - I)

1 (b), 2 (a), 3 (b), 4 (a), 5 (d), 6 (b), 7 (d), 8 (d), 9 (b), 10 (a), 11 (b), 12 (a), 13 (b), 14 (a), 15 (b), 16 (b), 17 (d), 18 (a), 19 (d), 20 (b), 21 (b)

CT303-N: Data Structures & Algorithms

Assignment # 7 (Tree - II)

1. 2-3 tree is a specific form of _____
(a) B tree
(b) B+ tree
(c) AVL tree
(d) Heap

2. Which of the following data structure can provide efficient searching of the elements?
(a) unordered lists
(b) binary search tree
(c) Heap
(d) 2-3 tree

3. Which of the following is NOT true about the 2-3 tree?
(a) all leaves are at the same level
(b) it is perfectly balanced
(c) postorder traversal yields elements in sorted order
(d) it is B-tree of order 3

4. Which of the following is false?
(a) 2-3 tree requires less storage than the BST
(b) lookup in 2-3 tree is more efficient than in BST
(c) 2-3 tree is shallower than BST
(d) 2-3 tree is a balanced tree

5. Which of the following is the most widely used external memory data structure?
(a) AVL tree
(b) B-tree
(c) 2-3 tree
(d) Both AVL tree and 2-3 tree

6. A B-tree of order 4 and of height 3 will have a maximum of ..... keys.
(a) 255
(b) 63
(c) 127
(d) 188

7. B-tree and AVL tree have the same worst case time complexity for insertion and deletion.
(a) True
(b) False

8. Consider a B+-tree in which the maximum number of keys in a node is 5. What is the minimum number of keys in any non-root node?
(a) 1
(b) 2
(c) 3
(d) 4

Answer Key -Assignment # 7 (Tree - II)

1 (a), 2 (d), 3 (c), 4 (a), 5 (b), 6 (a), 7 (a), 8 (b)

CT303-N: Data Structures & Algorithms

Assignment # 8 (Graphs)

1. In a simple graph, the number of edges is equal to twice the sum of the degrees of the vertices.
(a) True
(b) False

2. For a given graph G having v vertices and e edges which is connected and has no cycles, which of the following statements is true?
(a) v=e
(b) v = e+1
(c) v + 1 = e
(d) v = e-1

3. A graph with all vertices having equal degree is known as a _____
(a) Multi Graph
(b) Regular Graph
(c) Simple Graph
(d) Complete Graph

4. Which of the following ways can be used to represent a graph?
(a) Adjacency List
(b) Adjacency Matrix
(c) Adjacency List and Adjacency Matrix
(d) No way to represent

5. Which of the following is an advantage of adjacency list representation over adjacency matrix representation of a graph?
(a) In adjacency list representation, space is saved for sparse graphs.
(b) DFS and BSF can be done in O(V + E) time for adjacency list representation. These operations take O(V^2) time in adjacency matrix representation. Here is V and E are number of vertices and edges respectively.
(c) Adding a vertex in adjacency list representation is easier than adjacency matrix representation.
(d) All of the above

6. An adjacency matrix representation of graph cannot contain
(a) Nodes
(b) Edges
(c) Direction of edges
(d) Parallel edges

7. Which of the following statement(s) is/are true?
I. Adjacency list representation is better for sparse graph than adjacency matrix representation
II. Finding whether there is an edge between any two nodes in a graph is easier in Adjacency list representation
III. Adding a vertex in adjacency list representation is easier than adjacency matrix representation
(a) I only
(b) I & II only
(c) II & III only
(d) I II & III

8. An undirected graph G has n nodes. Its adjacency matrix is given by an n x n square matrix whose
I. Diagonal elements are '0'
II. Non-diagonal elements are '1'
Which of the following is true?
(a) Graph G has no minimum spanning tree (MST)
(b) Graph G has unique MST's each of cost n-1

(c) Graph G has multiple distinct MST's each of cost n-1
(d) Graph G has multiple spanning trees with different costs

9. How many distinct spanning trees do exist in an undirected cycle graph of n vertices?
(a) n
(b) n+1
(c) n-1
(d) n+2

10. If every node in a graph G is adjacent to equal number of nodes, then the graph is said to be
(a) Regular
(b) Finite
(c) Complete
(d) Strongly connected

11. The data structure required for Breadth First Search/Traversal on graph is
(a) Stack
(b) Queue
(c) Linked List
(d) Tree

12. The data structure required for Depth First Search/Traversal on graph is
(a) Stack
(b) Queue
(c) Linked List
(d) Tree

13. A connected (all vertices have at least one neighbor), undirected graph of N vertices has (N-1) edges. Number of spanning tree that can be constructed are
(a) 1
(b) N
(c) N-1
(d) N x N

14. Which of the following statement is always correct for any two spanning trees for a graph?
(a) Sum of weights of edges is always same
(b) Selected vertices have same degree
(c) Have same number of edges
(d) Have same number of edges and sum of weights of edges is also same

15. Which of the following statements is/are TRUE for an undirected graph?
I: Number of odd degree vertices is even
II: Sum of degrees of all vertices is even
(a) I Only
(b) II Only
(c) Both I and II
(d) Neither I nor II

16. For an undirected graph G with n vertices and e edges, the sum of the degrees of each vertex is
(a) nxe
(b) 2xn
(c) 2xe
(d) e^n

17. A complete graph can have ____
(a) n^2 spanning trees
(b) n^(n-2) spanning trees
(c) n^(n+1) spanning trees
(d) n^n spanning trees

18. Graph traversal is different from a tree traversal, because:
(a) trees are not connected
(b) graphs may have loops
(c) trees have root
(d) None of these

19. The number of edges in a simple, n-vertex, complete graph is
(a) n*(n-2)
(b) n*(n-1)
(c) n*(n-1)/2
(d) n*(n-1)*(n-2)

20. The spanning tree of connected graph with 10 vertices contains .....
(a) 9 edges
(b) 11 edges
(c) 10 edges
(d) 9 vertices

21. The number of edges in a regular graph of degree d and n vertices is
(a) nd
(b) n+d
(c) nd/2
(d) maximum of n,d

22. Which of the following properties does a simple graph not hold?
(a) Must be connected
(b) Must be unweighted
(c) Must have no loops or multiple edges
(d) All of the mentioned

Answer Key - Assignment # 8 (Graphs)

1 (b), 2 (b), 3 (b), 4 (c), 5 (d), 6 (d), 7 (b), 8 (c), 9 (a), 10 (a), 11 (b), 12 (a), 13 (a), 14 (c), 15 (c), 16 (c), 17 (b), 18 (c), 19 (c), 20 (a), 21 (c), 22 (a)

CT303-N: Data Structures & Algorithms

Laboratory Exercise # 6

(November 2 & 3, 2020)

Implement Chaining Method of Hashing Technique
[Source code: https://github.com/hbpatel1976/Data-Structure/blob/master/chaining.c]

Laboratory Exercise # 7

(November 9 & 10, 2020)

Implement (a) Linear Probing and (b) Quadratic Probing and (c) Double Hashing Methods of various Hashing Techniques

[Source Code:]
Linear Probing: https://github.com/hbpatel1976/Data-Structure/blob/master/linprob.c
Quadratic Probing: https://github.com/hbpatel1976/Data-Structure/blob/master/quadprob.c
Double Hashing: https://github.com/hbpatel1976/Data-Structure/blob/master/dblHashing.c

CT303-N: Data Structures & Algorithms

Assignment # 9 (Sorting and Searching)

1. Which of the following is not a stable sorting algorithm in its typical implementation?
(a) Insertion Sort
(b) Merge Sort
(c) Quick Sort
(d) Bubble Sort

2. Which of the following sorting algorithms in its typical implementation gives best performance when applied on an array which is sorted or almost sorted (maximum 1 or two elements are misplaced).
(a) Quick Sort
(b) Heap Sort
(c) Merge Sort
(d) Insertion Sort

3. Consider a situation where swap operation is very costly. Which of the following sorting algorithms should be preferred so that the number of swap operations are minimized in general?
(a) Heap Sort
(b) Selection Sort
(c) Insertion Sort
(d) Merge Sort

4. Suppose we are sorting an array of eight integers using quicksort, and we have just finished the first partitioning with the array looking like this: 2 5 1 7 9 12 11 10
(a) The pivot could be either the 7 or the 9.
(b) The pivot could be the 7, but it is not the 9
(c) The pivot is not the 7, but it could be the 9
(d) Neither the 7 nor the 9 is the pivot.

5. Suppose we are sorting an array of eight integers using heapsort, and we have just finished some heapify (either maxheapify or minheapify) operations. The array now looks like this: 16 14 15 10 12 27 28 How many heapify operations have been performed on root of heap?
(a) 1
(b) 2
(c) 3 or 4
(d) 5 or 6

6. In a selection sort of n elements, how many times is the swap function called in the complete execution of an algorithm?
(a) 1
(b) n^2
(c) n-1
(d) n x logn

7. Suppose that a selection sort of 100 items has completed 42 iterations of the main loop. How many items are now guaranteed to be in the final spot?
(a) 21
(b) 41
(c) 42
(d) 43

8. When is insertion sort is good choice for sorting an algorithm?
(a) Each component of the array requires large amount of memory

(b) The array has only few items out of space
(c) Each component of the array requires small amount of memory
(d) The processor speed is fast

9. The sorting machine can sort numbers in ascending orders only. This machine can be used for sorting all negative numbers in descending order by
(a) add 100 to all numbers, sort, subtract 100 from all numbers
(b) sort numbers, change sign of all numbers
(c) subtract the numbers from 100, sort, add 100 again to each number
(d) change sign of all numbers, sort numbers, and change sign again

10. If the number of records to be sorted is small, then …… sorting can be efficient.
(a) Merge
(b) Heap
(c) Selection
(d) Bubble

11. ____ is putting an element in the appropriate place in a sorted list yields a larger sorted order list.
(a) Insertion
(b) Extraction
(c) Selection
(d) Distribution

12. ____ is rearranging pairs of elements which are out of order, until no such pairs remain.
(a) Insertion
(b) Exchange
(c) Selection
(d) Distribution

13. Which of the following sorting algorithm is of divide and conquer type?
(a) Bubble sort
(b) Insertion sort
(c) Merge sort
(d) Selection sort

14. _____sorting algorithm is frequently used when n is small where n is total number of elements.
(a) Heap
(b) Insertion
(c) Bubble
(d) Quick

15. Which of the following sorting algorithm is of priority queue sorting type?
(a) Bubble sort
(b) Insertion sort
(c) Merge sort
(d) Selection sort

16. Which of the following is not an in-place sorting algorithm?
(a) Selection sort
(b) Heap sort
(c) Quick sort
(d) Merge sort

17. What is the advantage of bubble sort over other sorting techniques?
(a) It is faster
(b) Consumes less memory
(c) Detects whether the input is already sorted
(d) All of the mentioned

18. What is the worst-case time for serial search finding a single item in an array?
(a) constant time
(b) quadratic time
(c) logarithmic time
(d) linear time

19. What is the worst-case time for binary search finding a single item in an array?
(a) constant time
(b) quadratic time
(c) logarithmic time
(d) linear time

20. What additional requirement is placed on an array so that binary search may be used to locate an element?
(a) The array element must form a heap
(b) The array must have at least 2 entries
(c) The array must be sorted
(d) The array size must be a power of two

21. Which searching can be performed recursively?
(a) Linear
(b) Binary
(c) Both
(d) None

22. Which searching can be performed iteratively?
(a) Linear
(b) Binary
(c) Both
(d) None

23. The worst-case occur in linear search algorithm when …….
(a) Item is somewhere in the middle of the array
(b) Item is not in the array at all
(c) Item is the last element in the array
(d) Item is the last element in the array or item is not there at all

Answer Key - Assignment # 9 (Sorting and Searching)

1 (a), 2 (d), 3 (b), 4 (a), 5 (b), 6 (b), 7 (c), 8 (b), 9 (d), 10 (c), 11 (a), 12 (b), 13 (c), 14 (b), 15 (d), 16 (d), 17 (c), 18 (d), 19 (c), 20 (c), 21 (c), 22 (c), 23 (d)

CT303-N: Data Structures & Algorithms

Assignment # 10 (Hashing)

1. What is a hash function?
(a) A function has allocated memory to keys
(b) A function that computes the location of the key in the array
(c) A function that creates an array
(d) None of the mentioned

2. Hashing is the problem of finding an appropriate mapping of keys into addresses.
(a) True
(b) False

3. Which of the following is used in hash tables to determine the index of any input record?
(a) hash function
(b) hash chaining
(c) hash linked list
(d) hash tree

4. If several elements are competing for the same bucket in the hash table, what is it called?
(a) Diffusion
(b) Replication
(c) Collision
(d) None of the mentioned

5. What can be the techniques to avoid collision?
(a) Make the hash function appear random
(b) Use the chaining method
(c) Use uniform hashing
(d) All of the mentioned

6. Which of the following trait of a hash function is most desirable?
(a) It should cause less collision
(b) It should cause more collision
(c) It should be very easy to implement
(d) It should occupy less space

7. What is the load factor?
(a) Average array size
(b) Average key size
(c) Average chain length
(d) None of the mentioned

8. What is simple uniform hashing?
(a) Every element has equal probability of hashing into any of the slots
(b) A weighted probabilistic method is used to hash elements into the slots
(c) All of the mentioned
(d) None of the mentioned

9. A hash table of length 10 uses open addressing with hash function h(k)=k mod 10, and linear probing. After inserting 6 values into an empty hash table, the table is as shown below. Note that '_' denotes an empty location in the table.
0-_
1-_
2-42

3-23
4-34
5-52
6-46
7-33
8-_
9-_
Which one of the following choices gives a possible order in which the key values could have been inserted in the table?
(a) 46, 42, 34, 52, 23, 33
(b) 34, 42, 23, 52, 33, 46
(c) 46, 34, 42, 23, 52, 33
(d) 42, 46, 33, 23, 34, 52

10. How many different insertion sequences of the key values using the hash function h(k) = k mod 10 and linear probing will result in the hash table shown below? Note that '_' denotes an empty location in the table.
0-_
1-_
2-42
3-23
4-34
5-52
6-46
7-33
8-_
9-_
(a) 10
(b) 20
(c) 30
(d) 40

11. Consider a hash table of size seven, with starting index zero, and a hash function (3x + 4)mod7. Assuming the hash table is initially empty, which of the following is the contents of the table when the sequence 1, 3, 8, 10 is inserted into the table using closed hashing? Note that '_' denotes an empty location in the table.
(a) 8, _, _, _, _, _, 10
(b) 1, 8, 10, _, _, _, 3
(c) 1, _, _, _, _, _,3
(d) 1, 10, 8, _, _, _, 3

12. Consider a hash table of size seven, with starting index zero, and a hash function (3x + 4)mod7. Assuming the hash table is initially empty, which of the following is the contents of the table when the sequence 1, 3, 8, 10 is inserted into the table using closed hashing? Note that '_' denotes an empty location in the table.
(a) 8, _, _, _, _, _, 10
(b) 1, 8, 10, _, _, _, 3
(c) 1, _, _, _, _, _,3
(d) 1, 10, 8, _, _, _, 3

13. Given the following input (4322, 1334, 1471, 9679, 1989, 6171, 6173, 4199) and the hash function x mod 10, which of the following statements are true?
i. 9679, 1989, 4199 hash to the same value
ii. 1471, 6171 hash to the same value

iii. All elements hash to the same value iv. Each element hashes to a different value
(a) i only
(b) ii only
(c) i and ii only
(d) iii or iv

14. Which one of the following hash functions on integers will distribute keys most uniformly over 10 buckets numbered 0 to 9 for i ranging from 0 to 2020?
(a) h(i) =i2 mod 10
(b) h(i) =i3 mod 10
(c) h(i) = (11 ∗ i2) mod 10
(d) h(i) = (12 ∗ i) mod 10

15. Which of the following scenarios leads to linear running time for a random search hit in a linear-probing hash table?
(a) All keys hash to same index
(b) All keys hash to different indices
(c) All keys hash to an even-numbered index
(d) All keys hash to different even-numbered indices

16. The hash function is H1(k) = k % 50. In the case of collision, the hash function used is H(k) = (H1(k) + M x H2(k)) % 50 where H1(k) = k % 50 and H2(k) = k % 20. M is initialized to 0 and is incremented by 1 each time a collision occurs. This could be categorized under which of the following collision detection techniques.
(a) quadratic probing
(b) linear probing
(c) chaining
(d) double hashing

17. A hash table may become full in the case when we use open addressing.
(a) True
(b) False

Answer Key - Assignment # 10 (Hashing)

1 (b), 2 (a), 3 (a), 4 (c), 5 (d), 6 (a), 7 (c), 8 (a), 9 (c), 10 (c), 11 (c), 12 (b), 13 (c), 14 (b), 15 (a), 16 (d), 17 (a)

CT303-N: Data Structures & Algorithms

Assignment # 11 (File Organization)

Files are logically partitioned into storage units of fixed length known as
(a) Sectors
(b) Tracks
(c) Segments
*(d) Blocks

An organized logical sequence of record is called
*(a) File
(b) Organization
(c) Scrubbing
(d) Sequencing

Allocation of certain bytes are made at the beginning of the file is knows as
(a) File initiator
*(b) File Header
(c) File initializer
(d) Header

An index is clustered, if
(a) it is on a set of fields that form a candidate key.
(b) it is on a set of fields that include the primary key.
*(c) the data records of the file are organized in the same order as the data entries of the index.
(d) the data records of the file are organized not in the same order as the data entries of the index.