Data structure

ALNRAO

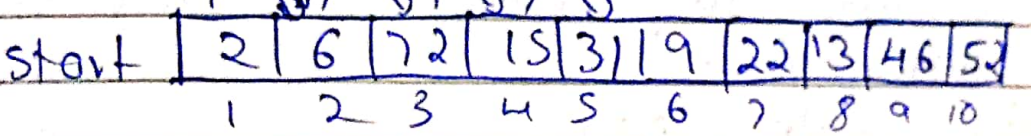
Unit - 1

Q1) What is linear search?

→ The simplest solution to the sequence search search problem is the sequential or linear search algorithm. This technique iterates over the sequence one item at a time until specific item is found or all the items are examined.

Example

a) Searching for 31



b) Searching for 6

Q2. Explain different operations on set data structure in python with example

1) A set is a container that stores a collection of unique values over a given comparable element.

2) Set() - create a new set initialized to the empty set

Ex: set = ("Raj")

3) Lenght() - returns the number of element in the set

Ex: print (len (set))

4) ADD() - set by adding given value to stored

Ex: - set.add("om")

Neeraj Appar
F-129
3/3

A.L.N.RAO

4) remove() - set by remove the given value or element to set
ex - set.remove("Om").
5) Union - (reate a new set that is union of the set and set
ex - set = set.union(set)
6) Interesect - new set is the intersection of the set and set.
def ex intersect (set)
7) difference (? - create and returns an iterator that can be used to iterate over collection of the
ex - difference (set)
8) Iterator - creates iterator
ex - it --- iter --- (set)

Q3 Write an algorithm for ~~intersection~~ insertion sort
-) Sorts are sequence in ascending order using insertion sorting algorithm

```
def intinsertionSort (theSeq):
    n = len (theSeq)
    #save the value to be positioned.
    value = theSeq(i)
    #find the position where value fits in ordered part of list
    pos = i
    while pos>0 and value < theSeq(pos-1)
    #shift the items to the right during search
    theSeq(pos) = theSeq(pos-1)
    pos = -1
    #put the saved value into open slot
    theSeq(pos) = value
```

Arrays Appav
1-129
3/3

pLN-RNO

Q4. Write an program to accept one name from user display whether the name exists in predefined name list.

```
-> s wnte = ["nv","om","ysh","schn"]
q = input ("enter a name)
if a in naive)
print (f"{a} is in names list.")
else
print (f"{a} is not in names list)
```

Output

enter a name - nv
nv is in names list

Q5 Write an algorithm for insertion sort. Discuss with example

```
-> def insertionSort (theSeq)
    n = len (theSeq)
    value = theSeq(i)
    pos = i
    while pos>0 and value< theSeq (pos-1)
    theSeq (pos) = theSeq (pos-1)
    pos-- = 1
    theSeq (pos) = value
```

Neeraj Appov.
I - 1st
31/3

A·L·N·RAO

Example

10 51 2 18 4 31 13 5 23 64 21

2 51 10 18 4 31 13 5 23 64 21

2 4 10 18 51 31 13 5 23 64 21

2 4 5 18 51 31 13 10 23 64 21

2 4 3 10 51 31 13 18 23 64 21

2 4 5 10 ~~8~~ 13 31 ~~51~~ 51 18 23 64 21

2 4 5 10 13 18 51 31 23 64 21

2 4 5 10 13 18 23 31 51 64 21

2 4 5 10 13 18 23 29 51 64 31

2 4 5 10 13 18 23 29 31 64 51

2 4 5 10 13 18 23 29 31 51 64

6) How to use list for maintaining sorted list

→ 1) To maintain a sorted list in real time new items inserted into their proper position

2) The new items cannot simply be appended at end of list as they may be out of order

3) Instead locate the proper position within the

Neeraj Appon
F-129
31/3

P.L.N.RAO

list and use the insert() method to insert in the indicated position.

iii) ~~The~~ To find the position of new item within a sorted list, a modified version of binary search can be used

5) The binary search uses a divide and conquer strategy to reduce number of items that may be examined: to find a target item or to determine the target is not in the list

1) How to implement multi arrays ADT in data structure

:) i) To implement the multi arrays ADT, the elements of the multi-dimensional array can be stored in a single 1-D array in row-major order

ii) Not only does this create a fast and compact array structure

3) The actual technique used by most programming language

4) A partial implementation of the multi-array class is provided

8) Write a python program to demonstrate binary search algorithm

```
def binarysearch(thevalues, target):
    # start with the entire sequence of elements
    low = 0
    high = len(thevalues) - 1
    # repeatedly subdivide the sequence in half
    until target is found
    while low <= high:
```

Name: Appan
1-129
31/3

A.L.N.RAO

of sequence

#Find the midpoint contain the de
mid = (high + low) // 2
#Does the midpoint contain the target
if the values (mid) == target:
return True.
#. or does the target precede the
midpoint)
elif target < the values (mid):
high = mid - 1
# or does it follow the midpoint
else:
low = mid + 1
# If the sequence cannot be subdi
further
return false

9) What are the various non-linear data structure
-) Non linear data structure where data ite
are not organised sequentially. A data
elements of non-linear data structure can
be connected to more than one element
to reflect a special relationship among
them. All the data elements in non lin
data structure can not be traversed in
single run.
The various type on non-linear data st
1) Tree - A tree is collection of nodes where
those nodes are arranged hierachically
and form a parent child relationship

Nirvraj Appan
F-120
31/3

Date: N RAS

a) Graph - A graph is a collection of finite number of vertices and or edges that connect those vertices Edges represent relationships among vertices that store data elements

10 ?) Sort the given list of numbers using insertion sort. show step by step procedure

14, 33, 27, 57, 100, 12

-) 14 33 27. 57 100 12

12 33 27 57 100 14

12 14 27 57. 100 33

12 14 27 33 100 57

12 14 27 33 57 100

Scanned with CamScanner

Neeraj Hpgari
F-129
04/04

Data structure

Unit II

A.L.N.RAO

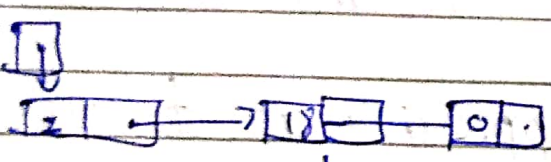
Q1) Write a python code to implement stack data structure operation using python list

```
stack = []
stack.append('a')
stack.append('b')
stack.append('c')
print(stack)
print('\n Element after pop')
print(stack.pop())
print(stack.pop())
print(stack.pop())
print(stack)
```

Q2 Write a short note on circular linked list

1) A singly linked list is a linked list in which each node contain a single link field and allows for a complete traversal from a distinctive find node to the last

eg



2) There are several operations that are commonly performed on a single linked list which we explore in their section

3) The circular linked list allows for a complete traversal of the nodes starting with any node in the list

4) It is commonly used for applications in which data is processed in round-robin fashion

Neeraj Appaw
F-129
04/04

A. L. N RAO

Q3) Give some applications of stack

a) i) Expression Evaluation
Stack is used to evaluate prefix, postfix and infix expression

ii) Expression Conversion
An expression can be represented in prefix, postfix or infix notation. Stack can be used to convert one form to expression to another

iii) Syntax Parsing
Many compilers use a stack for parsing the syntax of expressions, program blocks, etc. before translating into low level code.

iv) Parenthesis checking
Stack is used the proper opening and closing of parenthesis

v) Function call
Stack is used to keep information about the active functions or subroutines

Q4) How priority queue is implemented by using heap and hee

a) The standard approach it is to use an array starting at position 1, where each item in the array corresponds to one node corresponds to one node in the heap
i) The root of the heap is always in array[1]
ii) Its left child is in array[2]
iii) Its right child is in array[3]
iv) In general, if a node is in array[k], its left child is in array[k*2], and its

Neeraj Appaw
F-12a
04/04

A.L.Al-Rab

right child is in array [k*2+1]
v) If a node is in array [k], then its parent is in array [k/2]

Q5) Define function to put node at the end of the linked list.

-) def append (self, new_data):
    new_node = Node (new_data)
    if self.head != None:
        self.head = new_node
        return
    list = self.head
    while (loot.next):
        loot = loot.next
    loot.next = new_node.

Q6 Write a short note on singly link list

-) A) A linked list is, way to store a collection of elements like an array these can be character or integers each element in a linked list stored in the form of a node.
2) A linked list in a singly link field allows for a complete traversal then a distinctive first node to last.

Name: Appaji
I-124
04/04

A. NRAO

Q7) How queue is implemented using Python list

```
queue=[]
queue.append('a')
queue.append('6')
print(queue)
print("After removing")
print(queue.pop())
print(queue.pop())
print(queue)
```

Q8) Write a practical application of queue datastructure

-) A queue is a specialized list with a limited number of operation in which items can only be added to one end and removed from the other. A queue is known as a first-in-first-out (FIFO) list

Practical applications

1) The line of people waiting to be served like those would encounter at many business establishment. Each person is based on their position within the queue. Thus, the next person to be served is in first line

2) A queue datastructure is well suited for problems in data structure that requires data to be processed in the order which it was received

3) Queue are used in case of printer or uploading images, where the first one to be entered is first in process

4) In real life, call center phone system puts

Nirmal Appari
F-120
04101

D.L.N.Rao

Q9) Write an algorithm to evaluate expression using stack)

→ 1) Stack is a data structure used to store data in such a way that element inserted into the stack will be removed at last. Stack is first in last out structure.

2) The postfix notation is used to represent algebraic expressions. The expression written in postfix form are evaluated faster compared to infix notation as parenthesis are not required in postfix

3) Algorithm for postfix expression

1) Create a stack to store operands

2) If the element is a number, push it into the stack

3) If the element is a operator, pop operands for the operator from stack. Evaluate the operator and push the result back to stack

4) When the expression is ended, the number in the stack is final answer

Q10) Convert following infix to postfix

1) (A+B)/(C*D)

AB+/CD*

(AB+CD*)/

2) D*(E-F)/A+B

D*(EF-)/A+B

D*(EF-)/AB+

D(EF-)*/AB+

D(EF-)*AB+/

Noera's Appari
E-128
06/04

UNIT III - Data Structure

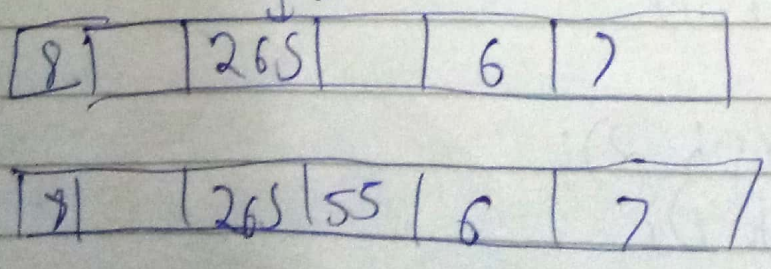
3) Write a python program to find sum of n number using recursive function

```
def sum(n)
    if n<=1
        return n
    else:
        return n+ sum(n-1)
num=int(input("Enter a number:"))
print("The sum is:",sum(sum))
```

output
Enter number - 10
The sum is 55

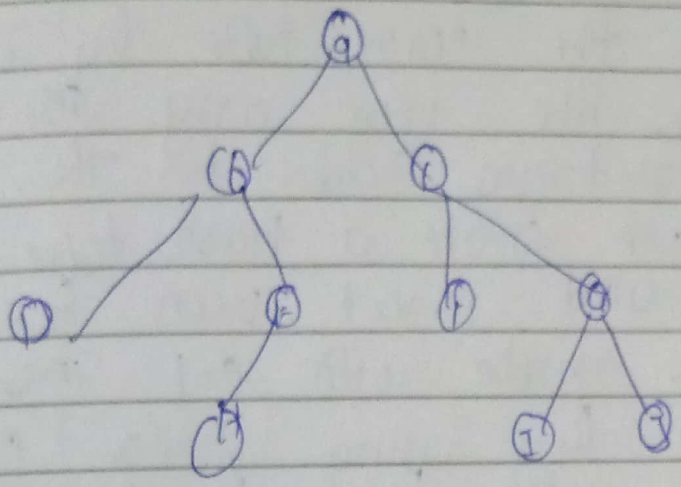
2) What do you meaning by hashing linear probing

Suppose the collision occurs the key have some hash location then we use linear probe. If two key map to the same table entry we must resolve collision by pushing the table to find another available slot with

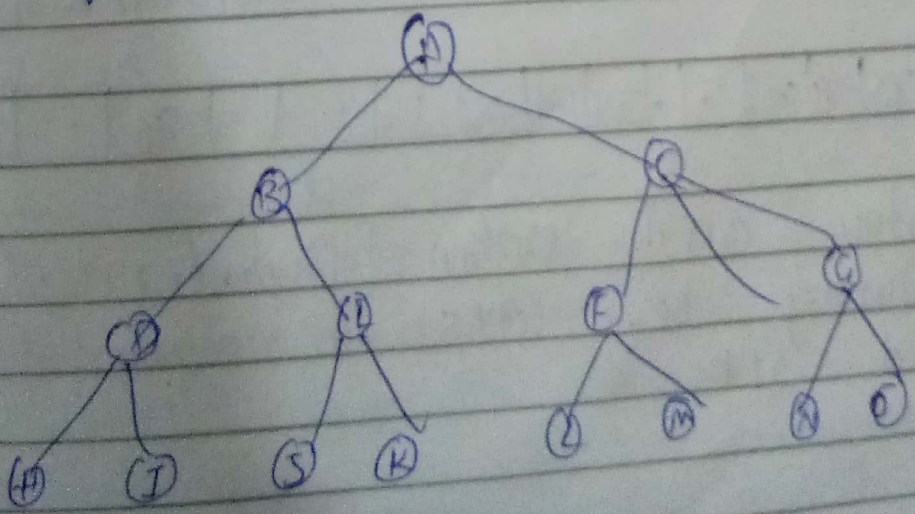


Niveraj Appari
F-129
06/04

Q3 Explain post order traversal with example.
→ In this traversal we start from the left side of the tree then to the right in the end we have root node.



Q4 Explain binary search tree?
→ Binary tree is a tree in which each node ~~can have~~ ~~but~~ most ~~two~~ two children one child is identified as the left child and on the other as a right child



Neeraj Appavi
F-129
06/04

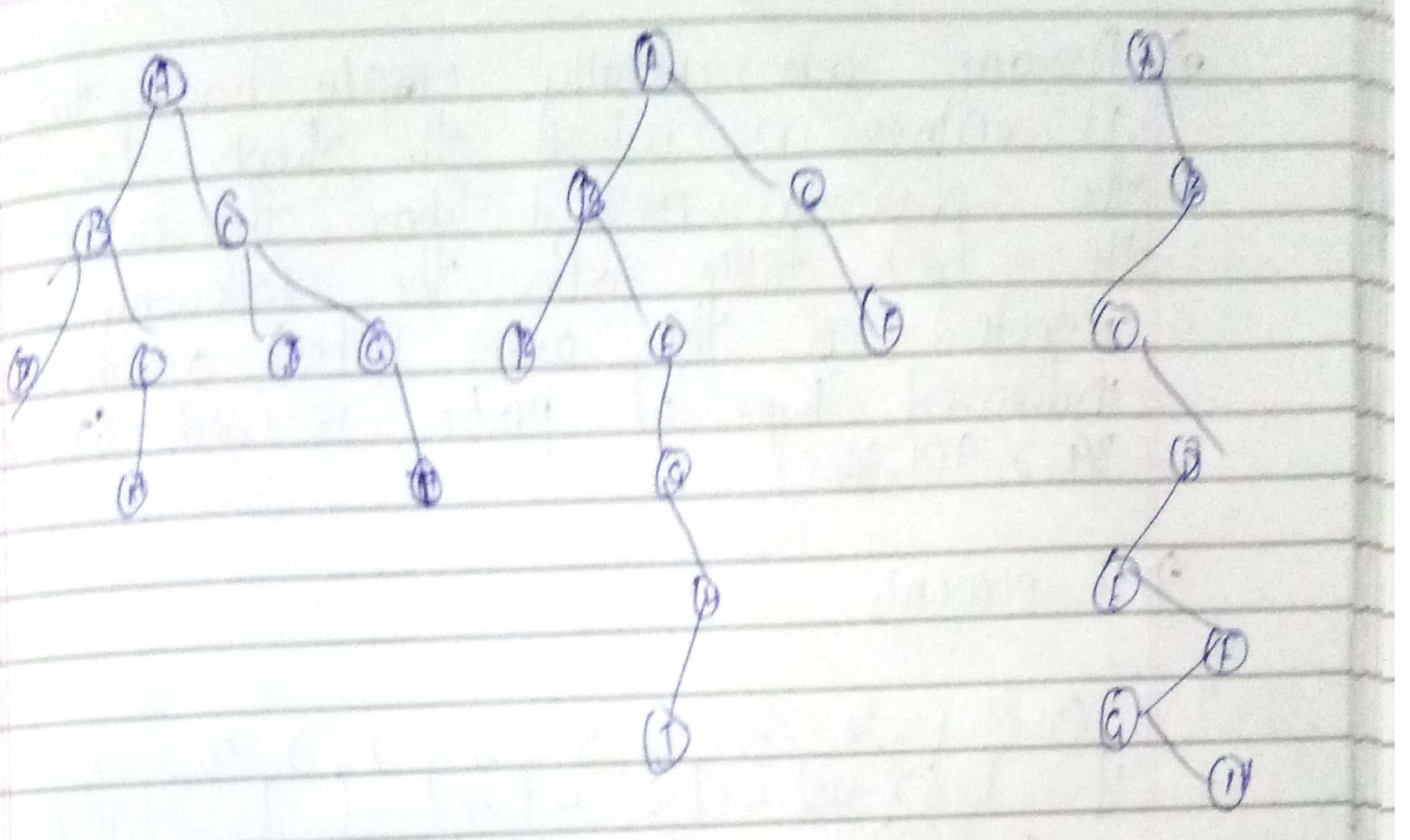
5) What is rehashing? Explain the example:
→ With hash table we can create a new array bigger than the original but we cannot simply copy the content from the old array to the new one. Instead we have to rebuild or refresh the entire table by adding each key to the new array as if it were a new key beginning addition. For the first for example we create a hash table of size 17 and insert given set of sample key using linear probe with c=1 insert given set of sample key using linear probe with c=1. Applying the hash function to the keys yields the following results

h(78) => 0
h(431) => 6
h(96) => 11
h(142) => 6 => 7
h(579) => 1
h(226) => 5
h(903) => 2
h(958) => 14

a) | 78 | 1 | 431 | | | | 96 | 226 | 579 | 903 | | | 265 | 142 |

b) | 265 | 829 | 903 | | | 226 | 431 | 142 | | | | 96 | | | 288 | | |

6) State and explain properties of hash
→ Binary hash comes in many different shapes and size



- Tree size - The size of the binary tree is to the number of nodes in the tree
Depth - It is the distance from the lit
height - It is the number of level in the tree
width - The number of nodes in a total containing the most node

Tree structure

Full binary tree. It is a tree in which each internal node containing two children

Tree Perfect binary tree - It is a tree in which all leaf nodes are at the same level Top to bottom with no gaps complete binary tree. A perfect binary tree down the height h-1 and node in the lowest level fill the available slots from left to right leaving touching the gap

Neeraj Appan
F-129
00/04

7) Assume an initially empty hash table with 11 entries in which the hash function uses the division method. Show the contents of the hash table after the following keys are inserted (in the order listed), assuming the indicated type of probe is used: 67, 8154, 39, 2, 901, 34

-> 11 elements

0	1	2	3	4	5	6	7	8	9	10
	67	815	45	2	34	39				90

h(67) => 1
h(815) => 1
h(45) => 1
h(39) => 6
h(2) => 2
h(901) => 10
h(34) => 1

8) Write a python code to find the factorial of a number using recursion

```
-> def factorial (n)
       if (n<=1):
           return 1
       else:
           return n* factorial (n-1)
num=5
print ("The factorial of 5:", factorial (num))
```

output - The factorial of 5:120

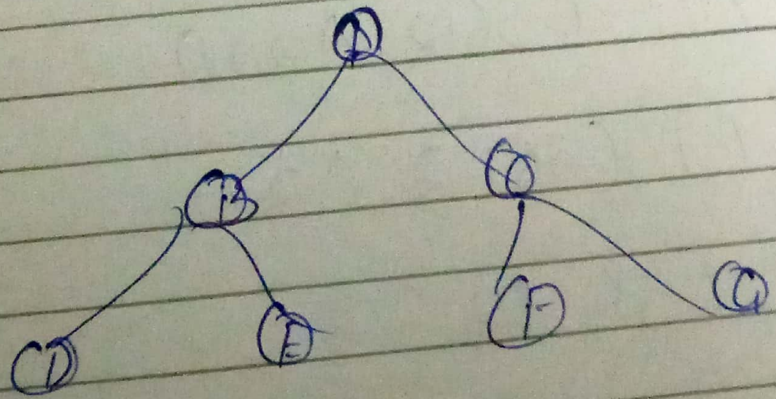
Neeraj Appari
f-129
06/04

A.L.N.RAO

9) Define search tree. Explain Binary search tree with example.

The tree structure consist of nodes and edges that organize data in a hierarchical fashion. The relationship between the data elements tree and family tree are similar. Only this tree contains data. Searching of these data in the tree is known as search tree.

• Binary search tree is a tree in which each node can have atmost two children one is identified as left child and another is identified as right child.



This is a binary tree

10) Convert infix into postfix

1) (a+b*c)-d/f

Neeraj Appan
I-129
0610

2) (-a+b) - 25 / 5 * 3+4

(-ab+) - 25 / 5 * 3+4

(-ab+) - 25 / 53 * +4

(-ab+) - 25 53 * 1+4

-ab+ - 25 53 * 14+

-ab+ 25 53 * 14+ -

3) (a/b*c) - 56 + 12^2

:) (a/b*c) - (56 + 12^2)

(a / bc*) - (56 12^2+)

(a bc* /) - (5612 ^2+1)

(ab c* /) (5612 ^2+) -

abc* / 56 12^2+ -