Assignment No :- 01

Title : write a program non Eecursive and non ~~dec~~recursive program to calculate fabonacces number.

objective : should to able to perform non- recurs program to calculate fabonacci number and analyse their time and space complexity

Prerrequisite : 1) Basic of python or java programming.

2) concept of recursive and non - recursive form

3) Basics of time and space complexity

Introduction :

introductive to fabonacci Series named after itrian mathematical pisana bogolla later known as fabonacci Search

What is fibonacci Series : The fabonacci series is the sequence of number is called fabbonaci search

Algorithm :-

STEP - I : Find the Smallest fabonacci number greater than or equal to n. let this number be fibm. two fabonacci no preceding it be fibmm1 and fibmm2.

STEP 2:- while the array has elements
be inspected

a) compare n with The last element of
range courred by fibmm2.

b) if x match, return index

c) else if x matches is less than th
element move the three fabonacci
variables two fabonacci down indic
elimination of approximately rear
Two - third of the removing array

d) else n is greater than the elemen
move The Three fabonacci variables
two fabonacci down

STEP 3 :- since there might be a sin
element remaning for compa
check if fibmm2 is 1-if yes com
n with thats remaning element i
match return index.

BC given as 0, 1, 1, 2, 3, 5, 8
21, 31. it can be observed that
term can be callculate.

given the terms f_0 and Second te
as 0 and 1 the third term

$$f_2 = 0 + 1 = 1$$

$f_3 = 1+1 = 2$
$f_4 = 2+1 = 3$

fabonacci sequence formula:

$f_0 = 0$ and $f_1 = 1$
$f_n = f_{n-1} + f_{n-2}$

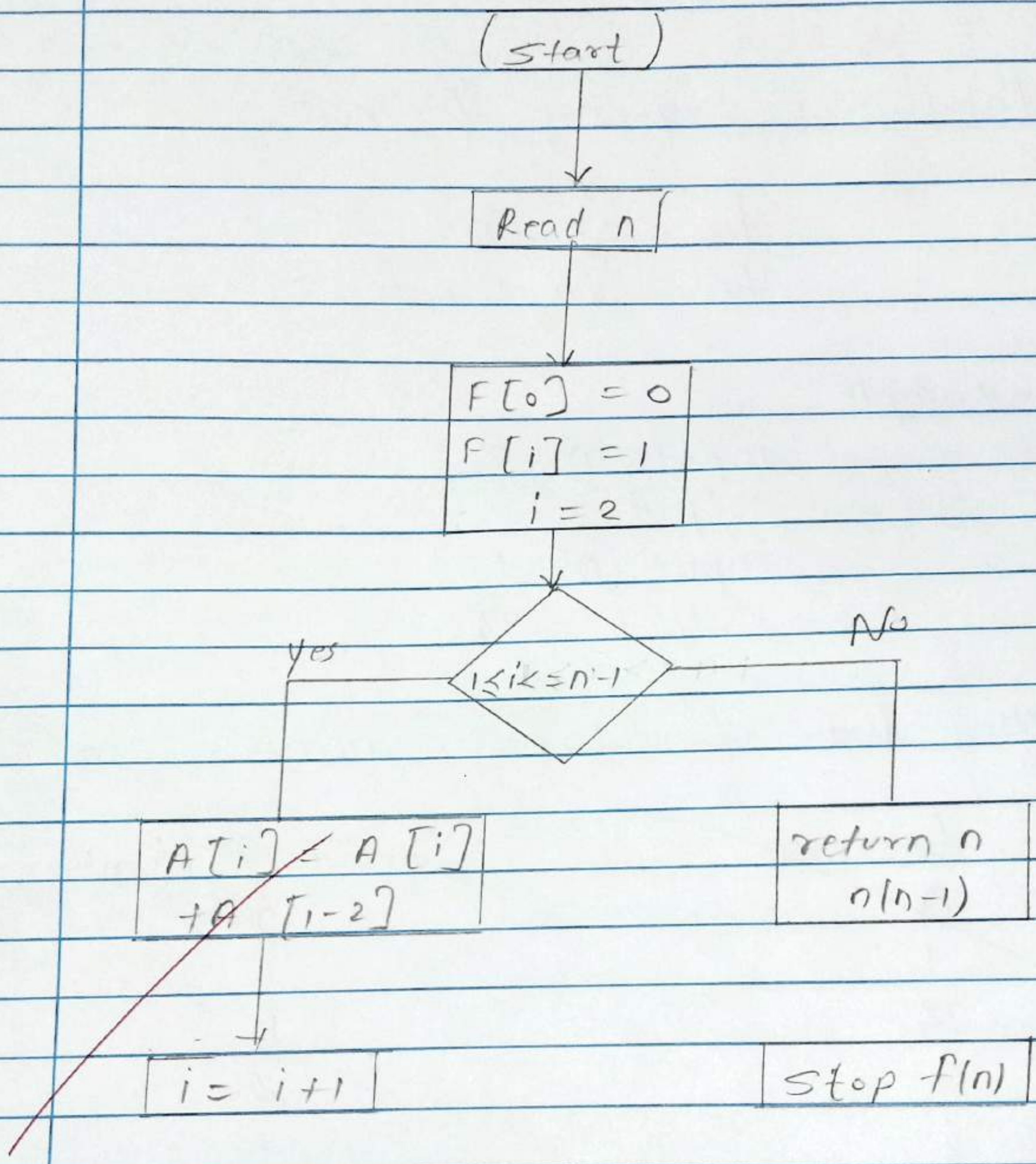
example:

input $n = 2$
output $= 1$
input $n = 9$
output $= 34$

the list of fabonacci number are calculated

fn	fabonacci number
0	0
1	1
2	1
3	2
4	3
5	5
6	8
7	13
8	21
9	34

flowchart :-



Application:

i) if is used to grouping of number and to study differents other special mathematics sequence

2) It is used in cryptography

Conclusion:

in this way have explored connept of fabonacci series using erecursire and non- recursire

Assignment No: 02

Title :- write a program to implement buffman encoding using stategy

Objective :- student should be able to solve Huffman encoding and greedy method

Prerequisite :-
1) Basic of python or Java
2) concept of greedy method.

Introduction :-
what is greedy method? A greedy method is an approoon for solving a program by selecting the best option avaliable at the movement it dosen't worry whether current best result.

Advantages of grredy approaches :-

1) This algorithm easier describe

2) This algorithm can perform better than other algorithm.

Disadvantages :-
1) It's doesn't produce the optimal
2) for example suppose we want to find longest path in the graph below root

to leaf

Huffman encoding:

1) Huffman coding is a tecnic of compres
data to eeduce it's size withod lo
of details it's was first develope
Hufman
2) Huffman coding is famous greedy algo
3) it is need to length enloding
4) it assign need to assign vanable len
code all characker

example:

A file contains the following ch
with the frequencies as shown in Hu
coding is used for data compression

Algorithm:

STEP - I

create a leaf node for each un
charater and buid min heap all th
~~m~~ nodes.

STEP - II

extract two nodes with the min
frequency from the min heap

STEP 3:-

create a new internal node with a frequency equal to the sum of the two nodes frequencies make the first extracted nodes as it's leaf child and the other extracte node as it's right child andd this node to the min heap.

STEP-4:-

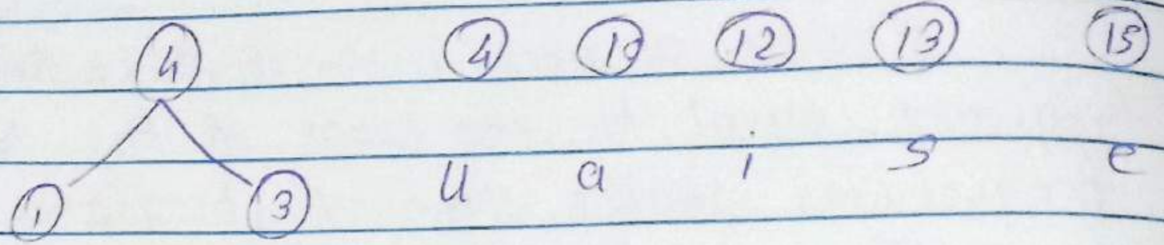
Repeat step2 and step3 until the heap contain any one node The remaining node is the root node and the tree is complete

character	frequencies
a	10
e	15
i	12
o	3
u	4
s	13
t	1

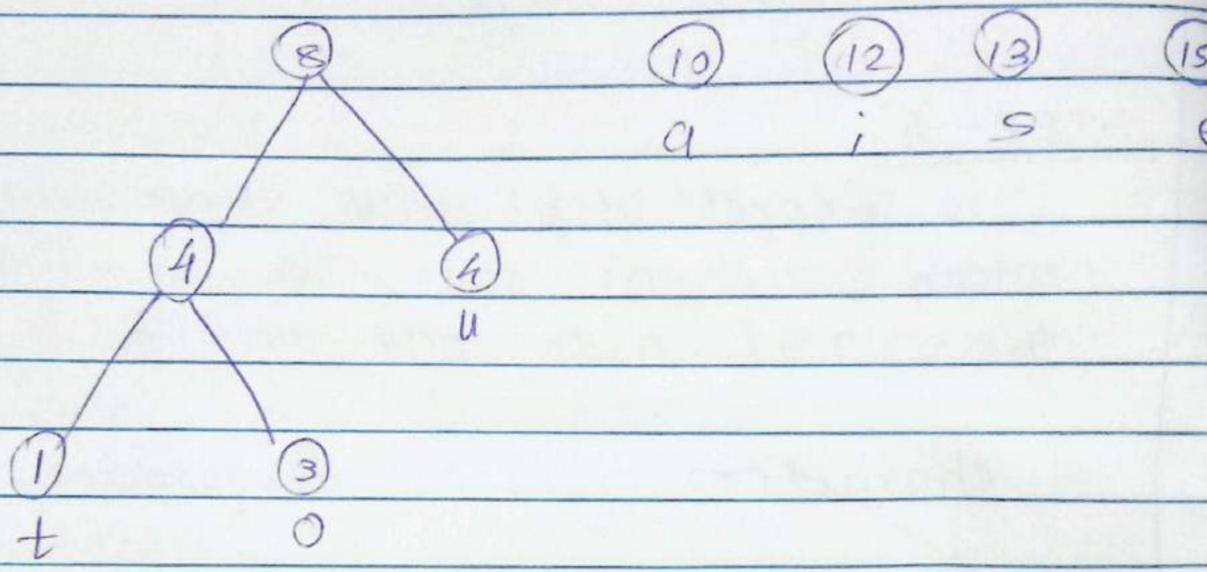
Step-1

1	3	4	10	12	13	15
t	o	u	a	i	s	e

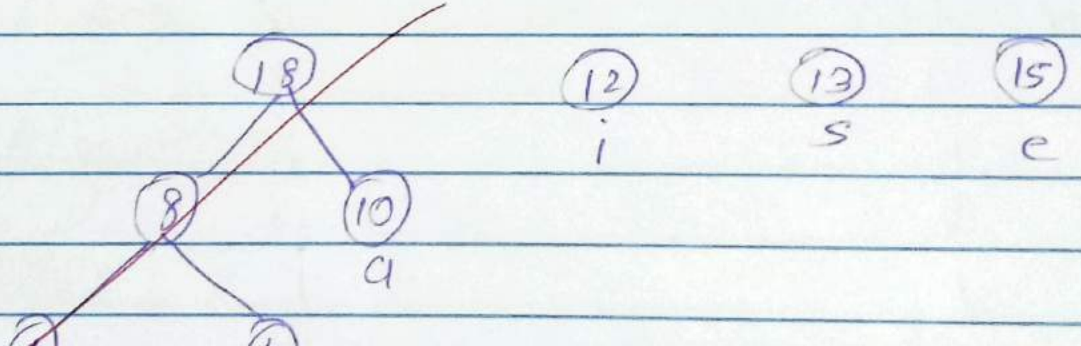
Step 2:


4
1
3
4
u
10
a
12
i
13
s
15
e

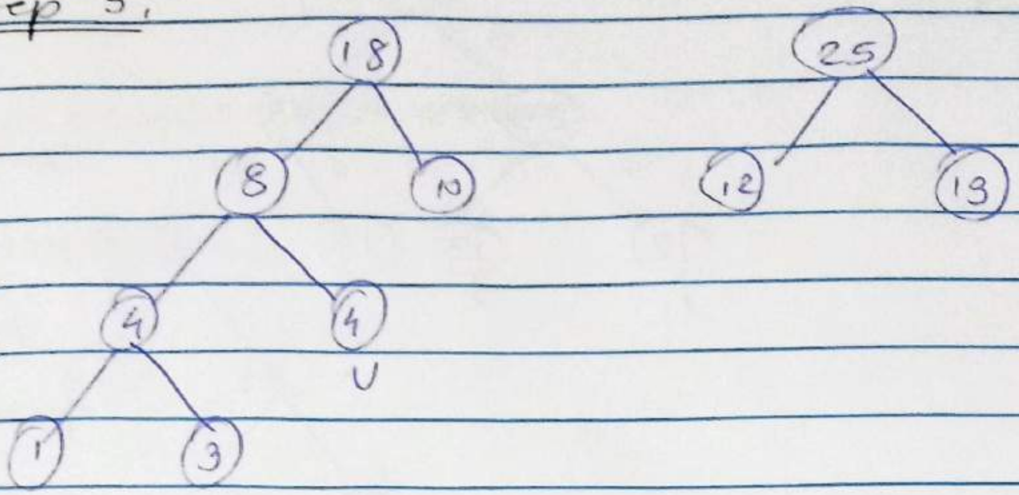
Step 3 :-


8
4
1
t
3
o
4
u
10
a
12
i
13
s
15

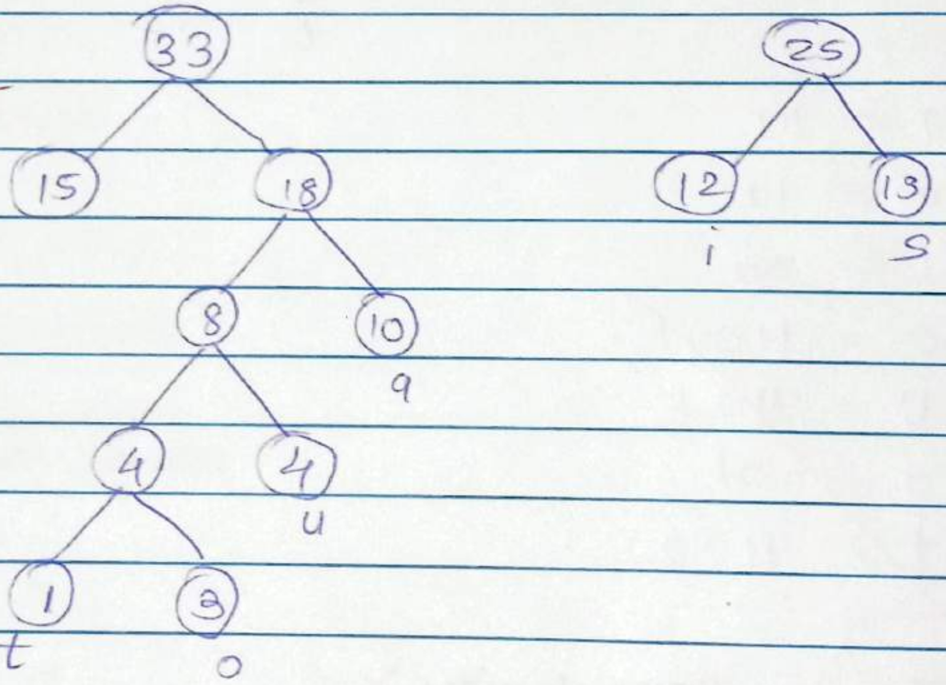
Step 4 :-


18
8
10
a
12
i
13
s
15
e

Step 5:-

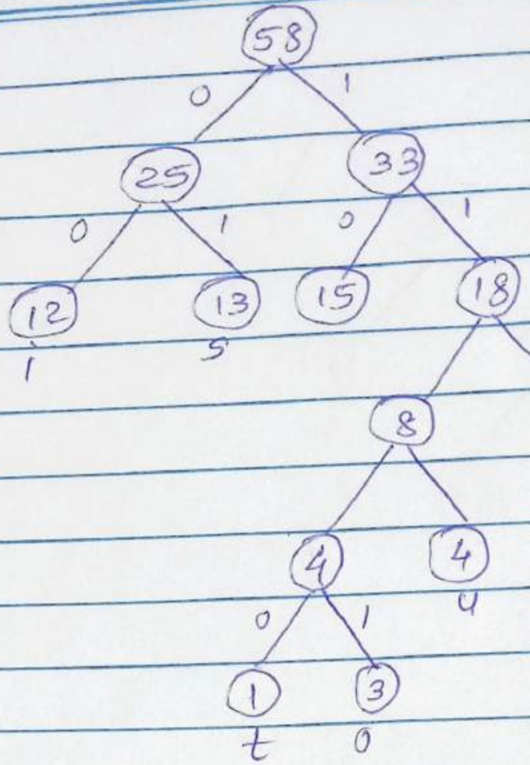


Step 6:-



Step 7:

step 7 :-



q = 111

e = 10

i = 00

o = 11001

u = 1101

s = 01

t = 11000

Time complexity :-

extract min() is called 2 x (n-1)

extract min() calls min (hepify)

Conclusion :- in this concept huffman

using greedy method

Bhide