27/09/2023

**PROGRAM EFFICIENCY:**

Efficiency is all about time and space complexity.

**:DYNAMIC PROGRAMING:**

**GREEDY APPROACH**: In greedy approach whatever is the solution for the problems has given at first go This is not the best approach for all the scenarios however it also works for some cases. In dynamic programming we will find out all the possible solution for the given problem out of which the best will be picked.

**Q.** Salim has 10000 in his account and annual interest is 12% in the 5th month he withdraw 25000 from his bank account and in the 9th month he deposit 10000 now calculate the final amount of his in financial year

COMPUND INTEREST:

ammount = 100000

annual\_roi = 0.12

monthly\_interest = annual\_roi/12

month\_in\_year = 12

withdrwal\_ammount = 25000

deposit = 10000

for month in range(4):

ammount += ammount \* monthly\_interest

ammount -= withdrwal\_ammount

for month in range(5, 8):

ammount += ammount \* monthly\_interest

ammount += deposit

for month in range(9, 12):

ammount += ammount\*monthly\_interest

ammount += ammount\*monthly\_interest

print(f"Swajith's account balance at the end of the financial year: {ammount:.2f} rupees")

SIMPLE INTEREST:

principal = 100000

annual\_interest\_rate = 0.12

withdrawal = 25000

deposit = 10000

SI1 = (principal \* 4 \* annual\_interest\_rate)/12

SI2 = ((principal - withdrawal) \* 4 \* annual\_interest\_rate)/12

SI3 = ((principal - withdrawal + deposit) \* 4 \* annual\_interest\_rate)/12

ammount = (principal - withdrawal + deposit)

ammount += SI1+SI2+SI3

print(f"Salim has ammount of in his account {ammount} rupess")

struc

{

Double

Char

}

16 byte /////////8+1+7

struc

{

Int

double

char

} //////24=4+4+8+1+7

**How many times statement is getting executed**

**Why? Efficiency of the program**

TYPE 1:

For ( i=0; i<n; i++) …….. n+1

{

Statements: …...... n+1

}

Polynomial therm f(n) = n+1

Which is O(n) big O of n or order of n

|  
20 x  
|

10 x  
|  
5 x

|-----10------20-------30--------🡪

#include<stdio.h>

int main()

{

int i, n=5;

for(i=0;i<n;i++)

printf("%d\n",i);

printf("final i :%d",i);

return 0;

} //////////////////////////////////Time complexity is = O(n);

for ( i=1; i<n; i+2)

{

Statements

}

n/2

f(n) = n/2

Degree of polynomial is n

So n/ anything is n So here also O(n)