Mon Tue Wed Thu Fri Sat (DAA) Date: 5/3/2024

Name: Affan Ahmad BS(cs) 5-A

Enrollment: 03-134221-003

Assignment: 1

Q(No1) Analysis:

Lets analyze loop step by step

1) initialization: The loop initialize i with n
2) condition: The loop continue as long as $i > 1$
3) update: In each iteration, i is updated to fun(i)
4) Body: The contain some O(1) expression

Worst case Analysis:

The worst case time is depend on the number of iteration the loop will perform

Lets assume that fun(i) reduces i by constant factor, say 'c'

In geometric form

$$n, \frac{n}{c}, \frac{n}{c^2} \ldots$$

let K be the number of iteration

$$\frac{n}{c^K} < 1$$

Mon Tue Wed Thu Fri Sat

Date: __/__/20__

Solving for k,

$$k > \log_c(n)$$

Therefore, the worst case time complexity is: $O(\log n)$

when fun(i) reduces i by a constant factor in each iteration

"(logrithm)"

(ii No2) The first loop has a time complexity of $O(\frac{m}{c})$

The second loop has a time complexity:

$$O(\frac{n}{c})$$

Worst case:

first The worst-case time complexity is $O(m/c)$ (assuming $c > 0$)

second The worst-case time complexity is $O(\frac{n}{c})$ (assuming $c > 0$)

"(Linear)"

Mon Tue Wed Thu Fri Sat Date __/__/20__

Qno3).) Basic operation:

-> comparison and swap inside the loop

*) loop Analysis :-

-> The outer loop runs 'n' times

-> n is the size of inpul array

-> Inner loop perform on the current iteration of outer loop.

*) Time complexity:

-> The inner loop perform $n.\frac{(n-1)}{2}$ in the worse case

*) Expressing complexity :-

-> The overall time complexity is $O(n^2)$ because the dominant term is n^2 "Cquadratic complexity)"