Analysis of Algorithms

Assignment 1

PROTIK DEY

Ans: to the Ques: No: 1(a)

Loop Invariant: After each iterations,

$$K = 2^c$$

For each iteration, k is multiplied by 2 and c is incremented by 1. That mean k gives 2^c for the current value of c in each iteration.

—

Ans: to the Ques: No: 1(b)

Base Case: Before the first iteration $K = 1$ and $C = 0$. According to the loop invariant,

$k = 2^c = 2^0 = 1$ which is the initial value of k. So the loop invariant holds.

Induction: Lets assume the loop invariant is true for first m iterations. So,

$$c = m \quad \text{and} \quad k = 2^m$$

After the next iteration,

$$K = 2^m \times 2 = 2^{m+1}$$

and $C = m+1$.

So the loop invariant holds for first $(m+1)$ iterations also.

At the end: The loop terminates when $C = n$. which gives us $K = 2^n$ which is the ~~dis~~ correct output of the program.

Therefore, we can say that the code correctly computes 2^n for any integer $n \geq 0$.
(Proved)

Ans: to the Ques: No: 1 (c)

The loop counter C starts at 0, runs as long as $C < N$ and increments by 1. So the run time of the algorithm is $O(n)$.

Ans to Ques No: 2(a)

The outer loop starts at $i = 3n$, runs till $i > 0$ and decreases by $i = i - 4$. So the number of iterations is:

(initial value - final value)/decrement

$\Rightarrow (3n - 0)\,4$

$\Rightarrow 3n/4$ which is $O(n)$

The inner loop starts at $j = 20n$, run till $j > 0$ and decreases by $j = j/3$. So the number of iterations is:

$\log_3 (20n)$

$\Rightarrow \log_2 (20n)$ which is $O(\log n)$.

Here both the loops runs for an exact number of times. So, this gives us a tight bound on both upper and ~~upper~~ lower bound. So here, total running time:

$O(n) \times O(\log n)$

$= O(n\log n) = \Omega(n \log n) = \Theta(n \log n)$

$\boxed{\text{Answer: } \Theta(n \log n)}$

Answer to Question No: 2(b)

The outer loop starts at $i = 3n^2$, runs till $i > 0$, and decreases by $i = i - 1$. So the number of iterations is: (initial value - final value) / decrement

$\Rightarrow \frac{3n^2 - 0}{1}$

$\Rightarrow 3n^2$ which is $O(n^2)$

The inner loop starts at $j = i$, runs till $j > 0$. and decreases by $j = j/2$. So for each value of i, the loops runs for $O(\log_2 i)$ times. Now i can be from $3n^2$ to 0, so the total run time for the inner loop is $\sum_{i=1}^{3n^2} \log_2 i$

$\approx n^2 \log_2 n$, which is $O(n^2 \log n)$.

Here, both the loops run for an exact number of iterations. So both the tight upper bound and tight lower bound is same.

So total running time:

$$O(n^2) \times O(n^2 \log n)$$

$$= O(n^2 \log n) = \Omega(n^2 \log n) = \Theta(n^2 \log n)$$

$$\boxed{\text{Answer: } \Theta(n^2 \log n)}$$

Ans: to the Ques: No: 3(a)

Here, $f(n) = 4n^5 - 50n^2 + 10n$ and $g(n) = n^5$

We have find constants c_1, c_2 and n_0 such that,

$$c_1 g(n) \le f(n) \le c_2 g(n) \text{ for all } c_1 > 0,\ c_2 > 0,\ n_0 > 0$$

$$\Rightarrow c_1 n^5 \le 4n^5 - 50n^2 + 10n \le c_2 n^5$$

Now,

$$\cancel{4n^5 - 50n^2 + 10n \le 4n^5 + 50n^5 + 10n^5 \text{ for}}$$

$$4n^5 - 50n^2 + 10n \le 4n^5 + 10n$$
$$\le 4n^5 + 10n^5$$
$$\le 15n^5 \text{ for all } n \ge 1$$

Here $c = 15$, $n_0 = 1$. So $f(n) = O(g(n))$.

Again,

$$3n^5 \leq 4n^5 - 50n^2 + 10n \text{ for all } n > n_0$$

and where n_0 is extremely large.

Here, n_0 is so large that the term $-50n^2+10n$ become negligible compared to the term $4n^5$

So, $c = 3$ and n_0 is large which is $n_0 > 0$.

So, $f(n) = \Omega(g(n))$

So we have $f(n) = O(g(n))$ ~~an~~ $= \Omega(g(n))$ .. Now ~~can~~ we can say that $f(n) \in \Theta(g(n))$

(Proved)

Ans: to the Ques: No: 3(b)

According to Limit Theorem, for two functions $f(n)$ and $g(n)$, where $g(n)$ is non-zero for sufficiently large n, if $\lim_{n\to\infty} \left| \frac{f(n)}{g(n)} \right| = 0$, then ~~$f(n) \in o(g(n))$~~

$$f(n) = o(g(n))$$

Here, $f(n) = 5n^{\frac{2}{3}} + 8\log n$, $g(n) = n$. We have to prove that, $\lim_{n\to\infty} \left| \frac{5n^{\frac{2}{3}} + 8\log n}{n} \right| = 0$

Here, if n becomes greater, then the term n grows faster than the other terms.

So when $n \to \infty$, then the equation approaches zero. So we can say that,

$$\lim_{n \to \infty} \frac{5n^{\frac{2}{3}} + 8\log n}{n} = 0.$$

So ~~we can~~ according to Limit Theory,

$$5n^{\frac{2}{3}} + 8\log n \in o(n)$$

(Proved)

Ans: to the Ques: No: 3(c)

Here; $f(n) = n^5 + 4n^2 + 15$ and

$g(n) = n^3$, we have to find a constant c

and n_0 such that

$f(n) \geq c\,g(n)$ for all $n \geq 0$ and $c > 0$.

Now,
$$n^5 + 4n^2 + 15 \geq n^5 \geq 1 \cdot n^3 \text{ for all } n \geq 1$$

Here, $c = 1$ and $n_0 = 1$.

So, we can say that

$$n^5 + 4n^2 + 15 \in \Omega(n^3)$$ (Proved)

Ans: to the Ques. No: 4

There are six functions. If we ordere them $f_1, f_2, f_3, \ldots$ such that, $f_1 \in O(f_2), f_2 \in O(f_3)$ etc,

$f_1 = \log_2 n$

$f_2 = n^{\frac{1}{3}}$

$f_3 = n^5$

$f_4 = 2^{\sqrt{\log_2 n}}$

$f_5 = 10^n$

$f_6 = n^n$

—