HW-2

Thursday, September 29, 2022 3:10 PM

Ans 1)

(a) Let $f(n) = 5n^3 + 2n^2 + 3n$

For O, $c.g(n) \geq f(n)$ for all $n \geq n_0$

1. $5n^3 + 2n^2 + 3n$
2. $5n^3 + 2n^2 + 3n^2 \geq 5n^3 + 2n^2 + 3n$
3. $5n^3 + 5n^2 \geq 5n^3 + 2n^2 + 3n^2 \geq 5n^2 + 2n^2 + 3n$
4. $5n^3 + 5n^3 \geq 5n^3 + 5n^2 \geq 5n^3 + 2n^2 + 3n^2 \geq 5n^2 + 2n^2 + 3n$
5. $10n^3 = 5n^3 + 5n^3 \geq 5n^3 + 5n^2 \geq 5n^3 + 2n^2 + 3n^2 \geq 5n^3 + 2n^2 + 3n$

Conclusion : $5n^3 + 2n^2 + 3n \leq 10n^3$

$\therefore f(n) = O(n^3)$ where $c = 10$ and $n_0 = 1$

(b) Let $f(n) = \sqrt{7n^2 + 2n - 8}$

For $\Theta(n)$, $c1.g(n) \leq f(n) \leq c2.g(n)$ for every $n \geq n_0$

1. $\sqrt{7n^2 + 2n - 8}$
2. $\sqrt{7n^2 + 2n - 8n} \leq \sqrt{7n^2 + 2n - 8} \leq \sqrt{7n^2 + 2n^2 - 8}$
3. $\sqrt{7n^2 + 2n^2 - 8n^2} \leq \sqrt{7n^2 + 2n - 8n} \leq \sqrt{7n^2 + 2n - 8} \leq \sqrt{7n^2 + 2n^2 - 8} \leq \sqrt{7n^2 + 2n^2}$
4. $n = \sqrt{n^2} \leq \sqrt{7n^2 - 6n^2} \leq \sqrt{7n^2 + 2n - 8n} \leq \sqrt{7n^2 + 2n - 8} \leq \sqrt{7n^2 + 2n^2 - 8} \leq \sqrt{7n^2 + 2n^2} = \sqrt{9n^2} = 3n$

Conclusion : $n \leq \sqrt{7n^2 + 2n - 8} \leq 3n$

$\therefore f(n) = \Theta(n)$ where $c1 = 1$, $c2 = 3$ and $n_0 = 1$

(c) $c1.f(n) \geq d(n)$ — (1)

$c2.g(n) \geq e(n)$ — (2)

(1) * (2)

$(c1\ f(n))(c2.g(n)) \geq d(n)e(n)$

$(c1.c2)(f(n)g(n)) \geq d(n)e(n)$

Let $c = c1.c2$

$\therefore c.(f(n)g(n)) \geq d(n)e(n)$ for $n \geq n_0$

where $c(f(n)g(n)) = O(f(n)g(n))$

Hence, $d(n)e(n) = O(f(n)g(n))$

Ans 2)

```python
def example1(lst):
    """Return the sum of the prefix sums of sequence S."""
    n = len(lst)              # O(1)
    total = 0                 # O(1)
    for j in range(n):        # O(n^2)
        for k in range(1+j):  # O(n)
            total += lst[k]
    return total              # O(1)
```

$0, 1, 2, 3, \ldots n$

0,	1
0, 1	2
0, 1, 2	3
0, 1, 2, 3	4
0, 1, 2, 3, 4	5
0, 1, 2, 3, 4, ... n	n

$n * n = n^2$

$\therefore$ $O(n^2)$ for example1(lst)

```python
def example2(lst):
    """Return the sum of the prefix sums of sequence S."""
    n = len(lst)           # O(1)
    prefix = 0             # O(1)
    total = 0              # O(1)
    for j in range(n):     # O(n)
        prefix += lst[j]   # O(1)
        total += prefix    # O(1)
    return total           # O(1)
```

$\therefore$ $O(n)$ for example2(lst).

```python
def example3(n):
    i = 1              # O(1)
    sum = 0            # O(1)
    while (i < n*n):   # O(log_2(n^2))
        i *= 2         # O(1)
        sum += i       # O(1)
    return sum         # O(1)
```

$\therefore$ $O(\log_2(n^2)) = O(2\log_2(n)) = O(\log_2(n))$ or $O(\log(n))$

for example3(n)

```python
def example4(n):
    i = n
    sum = 0
    while (i > 1):
        for j in range(i):
            sum += i*j
        i //= 2
    return sum
```

$O(1)$ — `i = n`, `sum = 0`

$\Theta(2n)$ — `while` loop

$O(1)$ — `sum += i*j`

$O(1)$ — `return sum`

For the inner loop,

$$n + \underbrace{\frac{n}{2} + \frac{n}{4} + \frac{n}{8} + \cdots}_{\text{This approximates to } n}$$

$n + n = 2n$

$\therefore\ \Theta(2n) = \Theta(n)$ for example4(n)