Take Home Assignment

Index No: ~~Sajeer~~ 210553J Name: Sajeenthiran Parameswaran

Q1) ~~Θ (B~~

o (little o)

$o(g(n)) = \{ f(n)$; for any positive constant $c > 0$, $\exists\, n_0 > 0$ s.t $0 \le f(n) < c\,g(n)$ ~~3~~ $\forall n > n_0 \}$

Example $2n = o(n^2)$

~~Let $c > 0$ be arbitary.~~ $c > 0$ be orbitary.

we know $2n \ge 0 \quad \forall n > 0$

$2n < c\,n^2$

$0 < cn^2 - 2n$

$0 < n(nc - 2)$

$n > 0$

$\therefore \quad nc - 2 > 0$

$n > \frac{2}{c} \qquad n_0 = \frac{2}{c} + 1$

~~$\therefore \exists n = \frac{2}{c} \ \forall c > 0$~~ s.t

since $c > 0$ is arbitary

$\forall c > 0 \quad \exists n_0 = \frac{2}{c} + 1$ s.t $\forall n > n_0$ s.t

$0 \le f(n) < c\,g(n)$

Q2) Big Omega (Ω)

$\Omega(g(n)) = \{ f(n)$: $\exists\, c$, $n_0 > 0$ s.t $0 \le c\,g(n) \le f(n)$ for $\forall n \ge n_0$

example: ~~$4n^2 + 100n + 500 = \Omega(n^2)$~~

$3n + 2 = \Omega(n)$

$3n + 2 \ge n$

$2n \ge n$

$n \ge 1 \qquad \therefore c = 1 \qquad n_0 = 1$

RICHARD

little Omega (ω)

$\omega(g(n)) = \{ f(n) : \forall c>0, \exists n_0 > 0$ s.t $0 \le c\,g(n) < f(n) \quad \forall n \ge n_0 \}$

example :

$$\frac{n^2}{2} = \omega(n)$$

let $c > 0$ be arbitary.

$\frac{n^2}{2} > 0 \quad n \ge 0$

$\cancel{cn} \le cn < \frac{n^2}{2}$

$0 < \frac{n^2}{2} - cn$

$0 < n\left(\frac{n}{2} - c\right)$

$n > 0 \quad \therefore$

$\frac{n}{2} - c > 0$

$\cancel{> c}$

$n > 2c$

$n_0 = 2c+1$

$\therefore$ since $c > 0$ ~~arbitar~~ arbitary

$\forall c > 0 \quad \exists n_0 = 2c+1$ s.t $\cancel{0 \le c\,g(n)} = 0 \le cn < \frac{n^2}{2}$

$\forall n \ge n_0$

Q3 Version 1

Code	Cost	Times.
for j = A.length to 2 do	c_1	n
swapped = false	c_2	$n-1$
for i = 2 to j do	c_3	$\frac{(n+1)n}{2} - 1$
swapped = false.	c_4	$n(n-1)/2$
if (A[i-1] > A[i]) then	c_5	$n(n-1)/2$
temp = A[i]	c_6	$n(n-1)/2$
A[i] = A[i-1]	c_7	$n(n-1)/2$
A[i-1] = temp	c_8	$n(n-1)/2$
swapped = true.	c_9	$n(n-1)/2$
if (! swapped) then	c_{10}	$n(n-1)/2$
break	c_{11}	0
n = newlimit	c_{12}	$n-1$

$$T(n) = c_1 n + c_2 (n-1) + c_3\left((n+1)(n)/2 - 1\right) \cancel{c_4 + c_5 + c_6 + c_7 + c_8 + c_9 +}$$
$$+ \left(c_4 + c_5 + c_6 + c_7 + c_8 + c_9 + c_{10}\right) n(n-1)/2 + c_{12}(n-1)$$

$$= c_{13} n + c_{14} n^2 + c_{15}$$

$$T_n = O(n^2)$$

Version 2

Code	Cost	Time
n = A.length	c_1	1
do	c_2	1
swapped = false	c_3	$(n+1)(n)/2$
for i = 2 to n do	c_4	$(n+1)(n)/2 - 1$
if (A[i-1]) > A[i] then	c_5	$(n-1)(n)/2$
temp = A[i]	c_6	$(n-1)(n)/2$
A[i] = A[i-1]	c_7	$(n-1)(n)/2$
A[i-1] = temp	c_8	$(n-1)(n)/2$
newlimit = i-1	c_9	$(n-1)(n)/2$
n = newlimit	c_{10}	$(n+1)(n)/2$
while swapped.	c_{11}	$(n+1)(n)/2$

$$T(n) = c_1 + c_2 + (c_3 + c_4 + c_{10} + c_{11})(n+1)(n)/2 + \cancel{c_5 + c_6 + c_7 + c_8 + c_9}$$
$$+ (c_5 + c_6 + c_7 + c_8 + c_9)(n-1)(n)/2.$$

$$= c_{10} + c_{11}(n+1)(n)\cancel{/2} + c_{12}(n-1)(n)\cancel{/2}$$

$$= c_{10} + c_{11}(n^2+n) + c_{12}(n^2-n)$$

$$T(n) = c_{10} + c_{13}(n^2) + c_{14}n$$

$\therefore$ $T(n) \in O(n^2)$ //

(2) No difference in time complexities.

Q (2)

Big oh (O)	little oh (o)	Big Omega (Ω)	little Omega (ω)	theta (Θ)
$O(g(n)) =$ ~~$\{f(n) : \exists c_0 > 0$~~ $\{f(n) : \exists c_0, n_0 > 0$ s.t $0 \leq f(n) \leq n g(n)$ $\forall n \geq n_0 \}$	$o(g(n)) =$ $\{f(n) : \forall c_0 > 0 \; \exists n_0 > 0$ s.t $0 ; 0 \leq f(n) < g(n) c \}$ $\forall n \geq n_0$	$\Omega(g(n)) =$ $\{f(n) ;$ $\exists c_0, n_0 > 0$ ~~$\forall c_0 > 0, \exists n_0 > 0$~~ ~~s.t $0 \leq f(n) \leq$~~ ~~s.t $0 \leq c g(n) \leq f(n) \leq$~~ $0 \leq c g(n) \leq f(n) \}$	$\omega(g(n)) =$ $\{f(n) ; \forall c_0 > 0 \; \exists n_0 > 0$ s.t $0 \leq g(n) c < f(n)$ $\forall n \geq n_0 \}$	$\Theta(g(n)) =$ $\{\exists c_1, c_2, n_0 > 0$ s.t $0 \leq c_1 g(n) \leq f(n) \leq$ $\forall n \geq n_0 \}$
Asymptotic upper bound	Asymptotic upper bound	Asymptotic lower bound	Asymptotic lower bound	Asymptocially tight bounds.
May or may not be a ~~tight~~ assymptotic tight bound	not assymptatically tight bound	May or may not be a assymptotical. tight lower bound	not assymptoticaly tight bound	

$$f(n) = O(g(n)) \Longleftrightarrow g(n) = \Omega(f(n))$$

$$f(n) = o(g(n)) \Longleftrightarrow g(n) = \omega(f(n))$$

$$f(n) = O(g(n)) \text{ and } f(n) = \Omega(g(n)) \Leftrightarrow f(n) = \Theta(g(n))$$

Yes, Instead of ~~taking~~ going through each line of the algorithm, we can only consider the loops.

We can find the no of times the loop statement runs.

If there are nested loops, then we can multiple them.

If there are several outer loops we can add them.

For example.

For $i = 1$ to n do — runs n times

For $j = 1$ to $\frac{n}{2}$ do — runs $\frac{n}{2}$ times.

----

for $k = 1$ to $\frac{n}{4}$ do — runs $n/4$ times

$\therefore$ Time taken is $O\left(n\frac{n}{2} + \frac{n}{4}\right)$

$O(n^2)$