Take home assignment.

210296X

Complexity Analysis.

Date No

Q1

1) Little Oh notation (o)

$o(g(n)) = \{ f(n) :$ for any positive constant $c>0$, there exists a constant $n_0 > 0$ such that $0 \leq f(n) < c\,g(n) \quad \forall n \geq n_0 \}$

We use o-notation to denote an upper bound that is not asyptotically tight.

For an example, $2n = o(n^2)$, but $2n^2 \neq o(n^2)$

Another kind of definition, $\lim_{n \to \infty} \frac{f(n)}{g(n)} = 0$

2) Big Omega notation (Ω)

$\Omega(g(n)) = \{ f(n) : \exists$ positive constant c and n_0; $0 \leq c\,g(n) \leq f(n) \quad \forall n \geq n_0 \}$

Ω - notation provides an asymptotic lower bound on a function.

Example $f(n) = n^2$, $g(n) = 2n-1$

For $c=1$ & $n=1$ $\quad 0 \leq g(n) \leq f(n)$

$\therefore f(n) = \Omega(g(n))$

It is defined to analyze best case time complexity of an algorithm.

3) Little Omega Notation (ω)

It is very similar to big Omega notation.

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

A loose lower bound is denoted by this.

Example, $\frac{n^2}{2} = \omega(n)$, but $\frac{n^2}{2} \neq \omega(n^2)$

Another relation between f(n) and g(n) is

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$$

Q2

Θ (Theta)	O (Big Oh)	o (Little oh)	Ω (Big omega)	ω (Little omega)
Asymptotic tight bound	Asymptotic upper bound. (May or may not be tight.)	Asymptotic upper bound (not tight)	Asymptotic lower bound (may or may not be tight.)	Asymptotic lower bound (not tight)
$\Theta g(n) = \{f(n):$ $\exists\, c_1, c_2, n_0 > 0$ such that $0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n)$ $\forall n \geq n_0,\ n_0 \in \mathbb{N}\}$	$O(g(n)) = \{f(n):$ $\exists\, c, n_0 > 0$ such that $0 \leq f(n) \leq c\, g(n)$ $\forall n \geq n_0,\ n_0 \in \mathbb{N}\}$	$o(g(n)) = \{f(n):$ $\forall c > 0, \exists n_0 > 0$ such that $0 \leq f(n) < c\, g(n)$ $\forall n \geq n_0,\ n_0 \in \mathbb{N}\}$	$\Omega(g(n)) = \{f(n):$ $\exists\, c, n_0 > 0$ such that $0 \leq c\, g(n) \leq f(n)$ $\forall n \geq n_0,\ n_0 \in \mathbb{N}\}$	$\omega(g(n)) = \{f(n):$ $\forall c > 0, \exists n_0 > 0$ such that $0 \leq c\, g(n) < f(n)$ $\forall n \geq n_0,\ n_0 \in \mathbb{N}\}$
Denotes exact time complexity	Denotes worst case time complexity	Denotes an approximated worst case time complexity	Denotes best case time complexity	Denotes an approximated best case time complexity.

Date

No

P3

1) Algorithm :

worst case

```
for j = A.length to 2 do
    swapped = false
    for i = 2 to j do
        ~~swapped = false~~
        if (A[i-1] > A[i]) then
            temp = A[i]
            A[i] = A[i-1]
            A[i-1] = temp
            swapped = true
    if (!swapped) then
        break;
```

Cost	Times
c_1	n
c_2	$n-1$
c_3	$\frac{n(n+1)}{2}$
$\cancel{c_4}$	
c_4	$\frac{n(n-1)}{2}$
c_5	$\frac{n(n-1)}{2}$
c_6	$\frac{n(n-1)}{2}$
c_7	$\frac{n(n-1)}{2}$
c_8	$\frac{n(n-1)}{2}$
c_9	$n-1$
c_{10}	0

$n = A.length$

$$\therefore f(n) = \left(\frac{c_3}{2} + \frac{c_4}{2} + \frac{c_5}{2} + \frac{c_6}{2} + \frac{c_7}{2} + \frac{c_8}{2}\right) n^2 +$$

$$\left(c_1 + c_2 + \frac{c_3}{2} - \frac{c_4}{2} - \frac{c_5}{2} - \frac{c_6}{2} - \frac{c_7}{2} - \frac{c_8}{2} + c_9\right) n$$

$$+ \left(\frac{c_3}{2} - c_2 - \frac{c_4}{2} - \frac{c_5}{2} - \frac{c_6}{2} - \frac{c_7}{2} - \frac{c_8}{2} - \frac{c_9}{2}\right)$$

$$= c' n^2 + c'' n + c'''$$

$$= O(n^2)$$

Date No

Algorithm 2 worst case :

Algorithm 2	Cost	Times
n= A.length	c_1	1
do	c_2	n
swapped = false	c_3	$n-1$
for i=2 to n	c_4	$\frac{n(n+1)}{2}$
if A[i-1] > A[i]	c_5	$\frac{n(n-1)}{2}$
temp = A[i]	c_6	$\frac{n(n-1)}{2}$
A[i] = A[i-1]	c_7	$\frac{n(n-1)}{2}$
A[i-1] = temp	c_8	$\frac{n(n-1)}{2}$
swapped = true	c_9	$\frac{n(n-1)}{2}$
newLimit = i-1	c_{10}	$\frac{n(n-1)}{2}$
n = newLimit	c_{11}	$n-1$
while swapped	c_{12}	n

$$f(n) = \left(\frac{c_4+c_5+c_6+c_7+c_8+c_9+c_{10}}{2}\right)n^2 +$$

$$\left(c_2+\frac{c_4}{2}-\frac{c_5}{2}-\frac{c_6}{2}-\frac{c_7}{2}-\frac{c_8}{2}-\frac{c_9}{2}-\frac{c_{10}}{2}+c_{11}+c_{12}\right)n$$

$$+\left(c_1-c_3+\frac{c_4}{2}-\frac{c_5}{2}-\frac{c_6}{2}-\frac{c_7}{2}-\frac{c_8}{2}-\frac{c_9}{2}-\frac{c_{10}}{2}-c_{11}\right)$$

$$= c_a n^2 + c_b n + c_c$$

$$= O(n^2)$$

2) As both versions have the same worst case complexity, there will be no difference.

3) Yes.
Without considering each and every step of the algorithm we can consider only the loops and their nested loops, and recursive terms. In this way we can calculate the worst case time complexity easily as we need the largest term of the function.