Question I: $A[1:70] = 11, 12, \ldots, 79, 80.$

iteration	size	# comparison
1	n	1
2	$\frac{n}{2}$	2
3	$\frac{n}{2^2}$	3
4	$\frac{n}{2^3}$	4
k	$\frac{n}{2^{k-1}}$	k

$n = 70$

So, number of comparison is the same of number of iteration of the loop.

for $x = 20$: it1 (iteration 1): size $= n \rightarrow 20 \neq$ mid1 ①

it2: size $= \frac{n}{2} \rightarrow 20 \neq$ mid2 ②

it3: size $= \frac{n}{2^2} \rightarrow 20 \neq$ mid3 ③

it4: size $= n/2^3 \rightarrow 20 \neq$ mid4 ④

it5: size $= n/2^4 \rightarrow 20 ==$ mid5 ⑤

Since number of iteration = 5, the number of comparison = 5

for $x = 45.5$: it1: size $= \frac{n}{2^0} \rightarrow 45.5 \neq$ mid1 ①

it2: size $= n/2^1 \rightarrow 45.5 \neq$ mid2 ②

it3: size $= n/2^2 \rightarrow 45.5 \neq$ mid3 ③

it4: size $= n/2^3 \rightarrow 45.5 \neq$ mid4 ④

it5: size $= n/2^4 \rightarrow 45.5 \neq$ mid5 ⑤

it6: size $= n/2^5 \rightarrow$ low == high, the loop will exit

Number of comparison $= 6 + 1 = 7 \rightarrow k = \log_2 n + 1$

for $x = 81$ it1: $s = n/2^0 \rightarrow 81 \neq$ mid1 ①

it2: $s = n/2^1 \rightarrow 81 \neq$ mid2 ②

it3: $s = n/2^2 \rightarrow 81 \neq$ mid3 ③

it4: $s = n/2^3 \rightarrow 81 \neq$ mid4 ④

//

// it6: $s = n/2^4 \rightarrow 81 \neq$ mid6 ⑥

Number of comparison $= 6 + 1 = 7 \rightarrow k = \log_2 n + 1$

Question II :. *Assuming n is even input.

① outer loop: $i : 0, 0+(1)(2), 0+2(2), 0+3(2), \ldots 0+r(2) = n \Rightarrow r = \frac{n}{2}$

$: 0, 1, 2, 3, r$

Therefore : $\sum_{r=0}^{n/2}$ inner loop .

Inner loop :

$j = i+1 \rightarrow j = 2r+1$

$$\sum_{j=i+1}^{n*n} 1 = \sum_{j=2r+1}^{n*n} 1 .$$

So

$$T(n) = \sum_{r=0}^{n/2} \sum_{j=2r+1}^{n*n} 1 = \sum_{r=0}^{n/2} n^2 - 2r - 1 + 1$$

$$= \sum_{r=0}^{n/2} n^2 - 2\sum_{r=0}^{n/2} r = n^2\left(\frac{n}{2} - 0 + 1\right) - \frac{\cancel{2}(n)(n+1)}{\cancel{4}}$$

$$= \frac{n^3}{2} + n^2 - n^2 - n$$

$$= \frac{n^3}{2} - n$$

---

② Since Statment 1 is the most expensive statment the cost will be $\theta(n^3)$

Question III :

we know that if $\lim_{n \to \infty} \frac{f(n)}{g(n)} = C$, where c is a non-zero constant, then, $f(n)$ is in $\Theta(g(n))$.

$f(n) = \log^2 n + n^{1/4}$, $g(n) = n^{\frac{1}{4}}$.

$$\lim_{n \to \infty} \frac{f(n)}{g(n)} = \lim_{n \to \infty} \frac{\ln^2 n + n^{1/4}}{n^{1/4}} = \lim_{n \to \infty} \frac{\ln^2 n}{n^{1/4}} + \lim_{n \to \infty} 1$$

$$= 1 + \lim_{n \to \infty} \frac{\ln^2 n}{n^{1/4}} \rightarrow H'R \quad = 1 + 0 = 1$$

Since 1 is a constant that is not equal to zero

$f(n)$ is in $\Theta(g(n)) \Rightarrow \log^2 n + n^{\frac{1}{4}}$ is in $\Theta(n^{\frac{1}{4}})$

Question IV :

① We know that the best case hapins with the lowest expinsive path. In Q4 code, the best case time complexty will be on line 4 " the return false statment after the if statment. If the two arrays sizes are equal time complexty will be $O(1)$.

② The worst case time complexty is when the two array don't equal each other. In this case the nested for-loop will be executed. The time complexty will be as follwes:

$$T(n) = \sum_{j=0}^{n-1} \sum_{k=0}^{m-1} 1 = \sum_{j=0}^{n-1} m = m(n) = mn$$

So, $O(mn)$ for the worst case. I choosed the return false statment insid the loop

③
int n = array1.length ; ①
int m = array2.length; ②
int j = 0 ③
int k = 0 ④

Space complexity : $\Theta(1)$

Question V :

① $f_1(n) = n\log n$, $f_2(n) = n$, $g(n) = n^2$

$f_1(n)$ is in $O(g(n)) \rightarrow n*\log n$ is in $O(n^2)$

$f_2(n)$ is in $O(g(n)) \rightarrow n$ is in $O(n^2)$

$f_1(n)$ is not in $O(f_2(n)) \rightarrow n\log n$ not in $O(n)$

②

1) False, exponitial functions does not garantee that the definition of big-Θ will be appicable. Example :

$$C_1 \log_2 n \leq n \leq C_2 \log_2 n \rightarrow \text{tak exponintial } 2$$

$$2^{f(n)} = 2^n \text{ , while } 2^{g(n)} = n$$

2) False, let $g(n) = n$, $f(n) = n^2 \rightarrow f(n)+g(n) = n^2+n$

$$C_1 n \leq n^2+n \leq C_2 n \quad ? \text{ wrong}$$

3) True, big-O-notiation allwos us to ignor the multibl constant by n, So 2^{an} is on $O(2^n)$

Q VI :

$f(n)$	$g(n)$	$f = O(g(n))$	$f = \Omega(g(n))$	$f = \Theta(g(n))$
$200n^3$	$100 + n + n^4$	True	False	False
$n \log n$	$\sqrt{n} \log^2 n$	False	True	False
2^n	3^n	True	False	False
n^n	3^n	False	True	False

Question VII :

1) for statment 1 : number of times $= \sum_{i=1}^{n-1} 1 = n-1-1+1 = n-1$

Statment 1 will be executed $(n-1)$ times wither $x=0$ or $x=10$ becuses it's indetendent of x.

for statment 2 :

outer loop: $i: 2^0, 2^1, 2^2, 2^3, \dots\ 2^r = n \to r = \log_2^n$

let $i = 2^r \to r = \log i$ and r rang from $\underline{0}$ to $\underline{\log_2^n}$

inner loop : $k: 1, 2, 3, \dots, i$; So ;

number of times statment 2 will be executed is :

$$\sum_{r=0}^{\log_2^n} \sum_{k=1}^{i} 1 = \sum_{r=0}^{\log_2^n} i = \sum_{r=0}^{\log_2^n} 2^r$$

$$= \frac{2^{\log_2^n+1} - 1}{2-1} = 2n - 1$$

if $x=0$ the statment won't be executed becuse of the if-statment

if $x=10 \to$ Statment 2 will be executed $(2n-1)$ times for statment 3 :

$i: 4^0, 4^1, 4^2, \ldots, 4^r$

$4^r = n^2 \rightarrow r = \log_4 n^2$

r ranges from 0 to $\log_4 n^2$

number of execuation for statment 3 :

$$\sum_{r=0}^{\log_4 n^2} 1 = \log_4 n^2 + 1$$

if $x = 0$, statment 3 will execute $(\log_4 n^2 + 1)$ times

if $x = 10$, statment 3 won't be execute anyway.

for statment 4 :

$i;\ 2^0, 2^1, 2^2, \ldots, 2^r = n \rightarrow r = \log_2 (n)$

outer loop: let $i = 2^r$ where r ranges from 0 to $\log_2 n$

inner loop: $k: 1, 2, 3, \ldots, i$

finally: $$\sum_{r=0}^{\log_2 (n)} \sum_{k=1}^{i} = \sum_{r=0}^{\log_2 n} i = \sum_{r=0}^{\log_2 n} 2^r$$

$$= \frac{2^{\log_2 n + 1} - 1}{2-1} = 2n - 1$$

I will add -1 because $i < n$ not $i <= n$

So, namber of execution is $2n - 1 - 1 = (2n-2) = 2(n-1)$

if $x = 0$ or $x = 10$ it will be $\cancel{(2n-1)}$ $2n-2$ because st 4 is independent of x.

~~2)~~ Statment 1 is $\Theta(n)$, Statment 2 is $\Theta(n)$
Statment 3 is $\Theta($~~logn~~ $\log_4 n^2)$, Statment 4 is $\Theta(n)$

in all cases st 1 and st 4 will be executed
but what about st2 and 3? if $x < 5$ st 3 will execute
but if $x \geq 5$ st 2 will execute. By takin the limit
we know that $f(St3) < g(St2)$ Which means that
St2 is more expensive than St3.

3) in the worst case, $\Theta\cancel{(\max(\Theta(St2), \Theta(St3)))} \cancel{= \Theta(St2)}$ $\Theta(\max(\Theta(st1), \Theta(st2), \Theta(st4)))$ $= \Theta(n)$

3) In the best case, $\Theta(\max(\Theta(St3), st(1), st(4))) = \Theta(n)$