Question1:

An ***algorithm*** is any well-defined computational procedure that takes some value, or set of values, as ***input*** and produces some value, or set of values, as ***output***. An algorithm is thus a sequence of computational steps that transform the input into the output.
The analysis of algorithm is the theoretical study of computer program performance and resource usage. And a particular focus on performance. **Algorithm design** on the other hand is a specific method to create a mathematical process in problem solving processes. Key properties include input, output, finiteness, definiteness, and effectiveness.

Question 2: Insertion Sort

5; 3; 4; 15; 0; 2; 8; 16
5; 3; 4; 15; 0; 2; 8; 16 // 3 not moved
5; 4; 3; 15; 0; 2; 8; 16 // Can show multiple steps or just direct is OK as well; 4
15, 5; 4; 3; 0; 2; 8; 16
15, 5; 4; 3; 0; 2; 8; 16
15, 5; 4; 3; 2; 0; 8; 16
15, 8; 5; 4; 3; 2; 0; 16
18;15; 8; 5; 4; 3; 2; 0

Book Page 25 for Loop Invariant Properties and Time Complexities

Question 3: Merge Sort

5; 3; 4; 15; 0; 2; 8; 16

5; 3; 4; 15	0; 2; 8; 16

5; 3	4; 15	0; 2	8; 16

5	3	4	15	0	2	8	16

5;3	15;4	2,0	16,8

15,5,4,3	16,8,2,0

16,15,8,5,4,3,2,0

Book Page 35 for Loop Invariant Properties and Time Complexities

$T(n) = 2T(n/2) + bn = O(n\log n)$

Question 4: Quick Sort

5; 3; 4; 15; 0; 2; 8; **16** // Pivot Point is 16

16,**3**,4,15,0,2,8,5 // 5 is replaced with 16; New pivot point is 3

16,4,15,8,**5**,*3*,**0**,2 // Now 5 & 0 i.e. before and after 3

16,15,**8**,5,**4**,3,**2**,0 // Now 2, 4, and 8

16,**15**,*8*,**5**,4,3,**2**,0 // Sorted.

Complexity O(n^2)

Question 5: n^2 + 50n + 6 = O(n^2)

f(n) = n^2 + 50n + 6
g(n) = cn^2

n^2 + 50n^2 +6n^2 = 57n^2
For c = 57, n_0 = 1; statement is true

Question 6: c = 13, n_0 = 2

Question 7: Give Full Marks if they write something about Big Oh, theta, and Omega

Question 8: (a) a = 7, b = 2, d =2
Case 3: $O(n^{\log_2 7})$

(b) a = 8, b = 2, d = 2

Case 3: $O(n^{\log_2 8})$_

(c) a = 4, b =2, d =1

Case 3: $O(n^{\log_2 4})$

Question 9:

$$T(n) = 4T\left(\frac{n}{2}\right) + n$$

$$\frac{T(n)}{n} = \frac{4T\left(\frac{n}{2}\right)}{n} + 1$$

$$\frac{T(n)}{n} = \frac{T\left(\frac{n}{2}\right)}{\left(\frac{n}{2^2}\right)} + 1$$

$$= \frac{T\left(\frac{n}{4}\right)}{\frac{n}{4^2}} + 1 + 1$$

$$\ldots \ldots$$

$$= \frac{T\left(\frac{n}{n}\right)}{\frac{n}{n^2}} + 1 + 1 + \cdots + 1$$

$$= T(1) \cdot n + \log n$$

$$= n^2 + n\log n$$

$$= O(n^2)$$

$$T(n) = 1 + T(n/2)\ ; \quad T(1) = \phi$$

$$= 1 + (1 + T(n/4))$$

$$= 1 + 1 + (1 + T(n/8))$$

$$= 1 + 1 + 1 + \cdots + 1 \ (1 + T(n/n))$$

$$= \log n + T(n/n)$$

$$= \log n + 1$$

$$T(n) = \log n$$

Question 10:

T **F** For all positive f(n), f(n) + o(f(n)) = Θ(f(n)).

Let f(n) = n^2

Then, n^2 + o(n^2) = Θ(n^2), For small o, f(n) < cg(n) i.e. o(n^2) should be less than n^2 . Thus, Equation is True

T **F** For all positive f(n), g(n) and h(n), if f(n) = O(g(n)) and f(n) = Ω(h(n)), then g(n) + h(n) = Ω (f(n))

f(n) = n, g(n) = nlogn, h(n) = logn, Thus, nlogn + logn = Ω(n), Thus Equation is True

T **F** If $f(n)= O(g(n))$ and $f(n)= \Omega (g(n))$, then we have $(f(n))^2 = \Theta ((g(n))^2)$

If $f(n) = 2n$, $g(n)$ can be n or $(2n-1)$ or any equation with linear n in order satisfy both $f(n)= O(g(n))$ and $f(n)= \Omega (g(n))$ simultaneously. Thus True

T **F** If $f(n) = O(g(n))$ and $f(n) = \Omega (g(n))$, then we have $f(n) = g(n)$

From above statement, it is clear that $f(n)$ and $g(n)$ can be different