Week 1 solution

(1) We know that $n \log n < n^2 < 2^n < n!$

Ans → (c) $f_3 < f_1 < f_4 < f_2$

(2) For ~~first~~ best case input, runtime of insertion sort algorithm is $O(n)$.

Ans → (a) linear.

(3) reverse sorted array represents the worst case time complexity in insertion sort which is $O(n^2)$

Ans → (b) array sorted in reverse order

(4) In insertion sort, after m passes through the array, the first m elements are not the m smallest element.

e.g. 4 2 1 0

5-1 2 4 1 0 (after 1 step) - (2 is not smallest in array)

Ans - (b)

(5)

Iteration 0	6	4	8	1	3
Iteration 1	4	6	8	1	3
Iteration 2	1	6	8	1	3
Iteration 3	(1)	4	6	8	3

$(25 \times 3) = 75$

Ans → (c) 75

(6) given array 27, 19, 33, 15, 1

I-1 19, 27, 33, 15, 1

I-2 19, 27, 33, 15, 1

I-3 15, 19, 27, 33, 1

ans → (b) 15, 19, 27, 33, 1

(7) $A(n) \leq W(n)$ for each case

$\therefore W(n) = c\,A(n)$ $c > 0$

$\therefore A(n) = O(W(n))$

Ans - (c) $A(n) = O(W(n))$

(8) worst case time complexity of merge sort is $O(n \log n)$.

Ans → (a) $O(n \log n)$

(9) no. of comparison needed to sort two list of size m and n (in the worst case) is $m + n - 1$

Answer → (d) $m + n - 1$

(10) When all the elements are identical

merge sort → $O(n \log n)$

insertion sort → $O(n)$

∴ insert sort will take least time

Ans → (b) false