AMCAT Searching and Sorting Questions

Question 1

What is recurrence for worst case of QuickSort and what is the time complexity in Worst case?

(A) Recurrence is T(n) = T(n-2) + O(n) and time complexity is O(n^2)

(B) Recurrence is T(n) = T(n-1) + O(n) and time complexity is O(n^2)

(C) Recurrence is T(n) = 2T(n/2) + O(n) and time complexity is O(nLogn)

(D) Recurrence is T(n) = T(n/10) + T(9n/10) + O(n) and time complexity is O(nLogn)

Answer: Option **(B)**

Explanation: The worst case of QuickSort occurs when the picked pivot is always one of the corner elements in sorted array. In worst case, QuickSort recursively calls one subproblem with size 0 and other subproblem with size (n-1). So recurrence is

T(n) = T(n-1) + T(0) + O(n)

The above expression can be rewritten as

T(n) = T(n-1) + O(n)

Question 2

Which of the following Sorting Algorithm will perform the worst if the numbers are ordered in the opposite form?

(A)Quick Sort

(B)Radix

(C)Bubble

(D)Selection

Answer: Option **A**

Explanation: Quick sort performs the worst if arranged in alphabetic/ ascending order

Question 3

Which of the following is not a stable sorting algorithm in its typical implementation.

(A) Insertion Sort

(B)Merge Sort

(C)Quick Sort

(D)Bubble Sort

Answer: Option **C**

Question 4

Binary Search can have _____ number of maxm comparsions?

(A)log(n) + 1

(B)2*log n

(C) n

(D)(n+1)/2

Answer: Option **A**

Explanation: Most number of comparision possible for BST is log(n) + 1

Question 5

Which of the following sorting algorithms in its typical implementation gives best performance when applied on an array which is sorted or almost sorted (maximum 1 or two elements are misplaced).

(A)Quick Sort

(B) Heap Sort

(C)Merge Sort

(D)Insertion Sort

Answer: Option **D**

Explanation: Insertion sort takes linear time when input array is sorted or almost sorted (maximum 1 or 2 elements are misplaced). All other sorting algorithms mentioned above will take more than lienear time in their typical implementation.

Question 6

What is the third number from the left while doing bubble sort in the 3rd iteration for 5 1 4 2 8?

(A) 4

(B) 5

(C) 2

(D) 8

Answer: Option **A**

Explanation: First Pass: (5 1 4 2 8) –> (1 5 4 2 8), Here, algorithm compares the first two elements, and swaps since 5 > 1. (1 5 4 2 8) –> (1 4 5 2 8), Swap since 5 > 4 (1 4 5 2 8) –> (1 4 2 5 8), Swap since 5 > 2 (1 4 2 5 8) –> (1 4 2 5 8), Now, since these elements are already in order (8 > 5), algorithm does not swap them. Second Pass: (1 4 2 5 8) –> (1 4 2 5 8) (1 4 2 5 8) –> (1 2 4 5 8), Swap since 4 > 2 (1 2 4 5 8) –> (1 2 4 5 8) (1 2 4 5 8) –> (1 2 4 5 8) Now, the array is already sorted, but our algorithm does not know if it is completed. The algorithm needs one whole pass without any swap to know it is sorted. Third Pass: (1 2 4 5 8) –> (1 2 4 5 8) (1 2 4 5 8) –> (1 2 4 5 8) (1 2 4 5 8) –> (1 2 4 5 8) (1 2 4 5 8) –> (1 2 4 5 8)

Question 7

Consider a situation where swap operation is very costly. Which of the following sorting algorithms should be preferred so that the number of swap operations are minimized in general?

(A)Heap Sort

(B)Selection Sort

(C)Insertion Sort

(D)Merge Sort

Answer: Option **B**

Explanation: Selection sort makes O(n) swaps which is minimum among all sorting algorithms mentioned above.

Question 8

Find the 3rd number from the left in the 3rd iteration while doing selection sort for – arr[] = 64 25 12 22 11

(A) 22

(B) 12
(C) 25
(D)64
Answer: Option **A**
Explanation: arr[] = 64 25 12 22 11 // Find the minimum element in arr[0…4] // and place it at beginning 11 25 12 22 64 // Find the minimum element in arr[1…4] // and place it at beginning of arr[1…4] 11 12 25 22 64 // Find the minimum element in arr[2…4] // and place it at beginning of arr[2…4] 11 12 22 25 64

Question 9
Which of the following is not true about comparison based sorting algorithms?
(A)The minimum possible time complexity of a comparison based sorting algorithm is O(nLogn) for a random input array
(B)Any comparison based sorting algorithm can be made stable by using position as a criteria when two elements are compared
(C)Counting Sort is not a comparison based sorting algorithm
(D)Heap Sort is not a comparison based sorting algorithm
Answer: Option **D**

Question 10
Which of the following algorithm will be the slowest amongst the following
(A) Shell
(B) Heap
(C)Quick
(D)Bubble
Answer: Option **D**