

## MCQs on Sorting Algorithms and Recursion

S No	Questions	Correct Answer
1.	<b>Which of the following sorting algorithms has the worst-case time complexity of <math>O(n^2)</math>?</b> a) Merge Sort b) Quick Sort c) Bubble Sort d) Insertion Sort	<i>c) Bubble Sort</i>
2.	<b>Which sorting algorithm repeatedly steps through the list, compares adjacent elements, and swaps them if they are in the wrong order?</b> a) Merge Sort b) Quick Sort c) Bubble Sort d) Insertion Sort	<i>c) Bubble Sort</i>
3.	<b>Which sorting algorithm is known for its average-case time complexity of <math>O(n \log n)</math> and worst-case time complexity of <math>O(n^2)</math>?</b> a) Merge Sort b) Quick Sort c) Bubble Sort d) Insertion Sort	<i>b) Quick Sort</i>
4.	<b>Which sorting algorithm works by repeatedly dividing the unsorted list into sublists and merging them back together?</b> a) Merge Sort b) Quick Sort c) Bubble Sort d) Insertion Sort	<i>a) Merge Sort</i>
5.	<b>Which sorting algorithm is considered stable, meaning it preserves the relative order of equal elements?</b> a) Merge Sort b) Quick Sort c) Bubble Sort d) Insertion Sort	<i>a) Merge Sort</i>
6.	<b>Which sorting algorithm is well-suited for sorting small arrays or nearly sorted arrays efficiently?</b> a) Merge Sort b) Quick Sort c) Bubble Sort d) Insertion Sort	<i>d) Insertion Sort</i>
7.	<b>Which sorting algorithm works by repeatedly selecting the minimum element from the unsorted portion of the array and moving it to the beginning?</b> a) Merge Sort	<i>c) Selection Sort</i>

## MCQs on Sorting Algorithms and Recursion

	b) Quick Sort c) Selection Sort d) Insertion Sort	
8.	<b>Which sorting algorithm is known for its simplicity and effectiveness for small data sets?</b> a) Merge Sort b) Quick Sort c) Bubble Sort d) Insertion Sort	<i>d) Insertion Sort</i>
9.	<b>Which sorting algorithm has a space complexity of <math>O(1)</math>, making it an in-place sorting algorithm?</b> a) Merge Sort b) Quick Sort c) Bubble Sort d) Insertion Sort	<i>c) Bubble Sort</i>
10.	<b>Which sorting algorithm works by partitioning the array into two partitions, sorting each recursively, and then combining them?</b> a) Merge Sort b) Quick Sort c) Bubble Sort d) Insertion Sort	<i>b) Quick Sort</i>
11.	<b>Suppose we are sorting an array of eight integers using quicksort, and we have just finished the first partitioning with the array looking like this:</b> <b>2 5 1 7 9 12 11 10</b>  <b>Which statement is correct?</b> a) The pivot could be either the 7 or the 9. b) The pivot could be the 7, but it is not the 9 c) The pivot is not the 7, but it could be the 9 d) Neither the 7 nor the 9 is the pivot.	a)The pivot could be either the 7 or the 9.
12.	<b>How many comparisons are needed to sort an array of length 5 if a straight selection sort is used and array is already in the opposite order?</b> a) 1 b) 5 c) 10 d) 20	c) 10
13.	<b>The number of swappings needed to sort the numbers 8, 22, 7, 9, 31, 5, 13 in ascending order, using bubble sort is</b> a) 11	1: 8, 7, 9, 22, 5, 13, 31 = 4 swaps 2: 7, 8, 9, 5, 13, 22, 31 = 3 swaps

## MCQs on Sorting Algorithms and Recursion

	<p>b) 12 c) 13 d) 10</p>	<p>3: 7, 8, 5, 9, 13, 22, 31 = 1 swap 4: 7, 5, 8, 9, 13, 22, 31 = 1 swap 5: 5, 7, 8, 9, 13, 22, 31 = 1 swap</p> <p>Total 10 swaps are required to sort the array. Hence Option(D) is the correct answer.</p>
14.	<p><b>If one uses straight two-way merge sort algorithm to sort the following elements in ascending order: 20, 47, 15, 8, 9, 4, 40, 30, 12, 17 then the order of these elements after second pass of the algorithm is:</b></p> <p>a) 8, 9, 15, 20, 47, 4, 12, 17, 30, 40 b) 8, 15, 20, 47, 4, 9, 30, 40, 12, 17 c) 15, 20, 47, 4, 8, 9, 12, 30, 40, 17 d) 4, 8, 9, 15, 20, 47, 12, 17, 30, 40</p>	<p>b) 8, 15, 20, 47, 4, 9, 30, 40, 12, 17</p>
15.	<p>Predict output of following program</p> <pre>#include &lt;stdio.h&gt; int fun(int n) {     if (n == 4)         return n;     else return 2*fun(n+1); } int main() {    printf("%d", fun(2));     return 0;}</pre> <p>a) 4 b) 8 c) 16 d) Runtime Error</p>	<p>c) 16</p>
16.	<p>Consider the following recursive function fun(x, y). What is the value of fun(4, 3)</p> <pre>int fun(int x, int y) {     if (x == 0)         return y;     return fun(x - 1, x + y); }</pre> <p>a) 13 b) 12 c) 9</p>	<p>a) 13</p>

## MCQs on Sorting Algorithms and Recursion

	d) 10	
17.	<p>Predict output of following program</p> <pre>void fun(int n) {     if (n == 0)         return;     printf("%d", n%2);     fun(n/2); }</pre> <p>a) 11001 b) 10011 c) 11111 d) 00000</p>	b> 10011

## MCQs on Sorting Algorithms and Recursion

18.	<pre>int fun(int x, int y) {     if (y == 0) return 0;     return (x + fun(x, y-1)); } int fun2(int a, int b) {     if (b == 0) return 1;     return fun(a, fun2(a, b-1)); }  a) x*y b) x+x*y c) xy d) yx</pre>	Answer:C    xy
-----	---	----------------