

<b>Rules to Answer</b> Accurate answer supported by evidence, and reasoning links answer and evidence. Response is specific and detailed. Presence of keywords and scientific terminology.	<b>Rules to Answer</b> Refer content if unable to recall after 15-30sec of trying, then again retrieve from memory without help.	<b>Recurrence</b> Main idea	<b>Recurrence</b> An equation that recursively defines a sequence once one or more initial terms are given: each further term of the sequence or array is defined as a function of the preceding terms.
<b>Recursion vs Recurrence</b> Explain	<b>Recursion vs Recurrence</b> A recurrence relation uses recursion to create a sequence. Recursion is not limited to generation of sequences.	<b>Recursion</b> Recursion is the repeated use of a procedure or action	<b>Recurrence</b> In fact, a recurrence relation uses recursion to define a sequence.
<b>Recurrence</b> This sequence is built in such a way that each term is defined as a combination of previous terms. The generation of such a sequence is a requirement in the definition	<b>Recurrence</b> Recurrence is analyzed by telescoping	<b>Binary Search</b> Main Idea	<b>Binary Search</b> Search algorithm that finds the position of a target value within a sorted array
<b>Binary Search</b> Elements are already sorted in ascending order Divide and Conquer algorithm is used	<b>Binary Search</b> Element x is either in the left half of the array or in the right half or not there at all	<b>Binary Search</b> Compare x to middle element k, $x > k$ (x not in left half), $x < k$ (x not in right half)	<b>Running time of Binary Search</b> Explain
<b>Running time of Binary Search</b> $\lg_2 N$ binary search runs in $O(\lg N)$	<b>One word answer</b> In computer science you use $\lg_2 N$ or $\lg_{10} N$	In computer science you use $\lg_2 N$ or $\lg_{10} N$ $\lg_2 N$	<b>Merging</b> Elaborate
<b>Merging</b> Merging is not a divide and conquer algorithm, but part of mergesort algorithm	<b>Merge Sort</b> Merge Sort is a divide and conquer algorithm	<b>Merging</b> Take two sorted arrays of numbers and make a single array which is sorted of all of those numbers	<b>Merge Sort</b> Type of Algorithm
<b>Merge Sort</b> Explain the complete algorithm	<b>Mergesort</b> First divide the list into smallest unit (1 element), then compare each element with the adjacent list to sort and merge two adjacent lists. Finally all the elements are sorted and merged.	<b>Run time for mergesort</b>	<b>Run time for mergesort</b> $O(n \cdot \lg n)$
<b>Insertion sort vs Mergesort</b> Which one is better and why?	<b>Insertion sort vs Mergesort</b> Insertion sort is $O(n^2)$ . Mergesort is $O(n \cdot \log n)$ . So, merge-sort is a superior algorithm in terms of its running time on large datasets.		