Prelude (Add emojis and make it interactive)

* SortViz
  + Hi 👋🏼, I’m Premnaath Varadharajan. This Playground helps learning different sorting algorithms using Augmented Reality.
* About
  + An algorithm is a finite sequence of well-defined instructions, typically used to solve a class of specific problem.
  + Sorting algorithms are prominently used in our daily life, from Apple Music’s sort Playlist feature to GitHub’s sort repositories option.
  + An efficient algorithm is the one that takes the least amount time and space.
  + Sometimes learning efficient sorting algorithms can be quite laborious.
* Learning Objectives
  + Bubble Sort
  + Insertion Sort
  + Merge Sort
  + Quick Sort
* Prerequisites
  + Complexity Analysis of Algorithms
  + Mindset to learn 😝
* NOTE
* Tech Stack for this page (if need be)
* Add audio/music while generating data set

Bubble Sort

* About
  + Bubble sort works on repeated swapping of adjacent elements until they are in the intended order.
  + It is called bubble sort because the movement of array elements is just like the movement of air bubbles in the water.
  + Bubbles in water rise up to the surface; similarly, the array elements in bubble sort move to the end in each iteration.
* Analysis
  + Time Complexity: O(n^2)
  + Space Complexity: O(1)
  + Stability: True
  + K-Passes: True

Insertion Sort

* About
  + Insertion sort works similar to the sorting of playing cards in hands.
  + It is assumed that the first card is already sorted in the card game, and then we select an unsorted card.
  + If the selected unsorted card is greater than the first card, it will be placed at the right side; otherwise, it will be placed at the left side.
  + Similarly, all unsorted cards are taken and put in their exact place.
* Analysis
  + Time Complexity: O(n^2)
  + Space Complexity: O(1)
  + Stability: True
  + K-Passes: False

Quick Sort

* About
  + Quicksort is the widely used sorting algorithm that makes (**n\*log n**) comparisons in average case for sorting an array of n elements.
  + It is a faster and highly efficient. This algorithm follows the divide and conquer approach.
  + Divide and conquer is a technique of breaking down the algorithms into subproblems, then solving the subproblems, and combining the results back together to solve the original problem.
* Analysis
  + Time Complexity: O(n\*log(n))
  + Space Complexity: O(log(n)) Recusrion Stack
  + Stability: False
  + K-Passes: False

Merge Sort

* About
  + Merge sort uses the divide and conquer approach to sort the elements.
  + It is one of the most popular and efficient sorting algorithm.
  + It divides the given list into two equal halves, calls itself for the two halves and then merges the two sorted halves.
* Analysis
  + Time Complexity: O(n\*log(n))
  + Space Complexity: O(n)
  + Stability: True
  + K-Passes: False

Quiz

* What makes apple devices fast?
  + High raw CPU processing power
  + Efficient algorithms and efficient System on Chip design (Correct)
  + Simple UI
  + Screen resolution
* Which sorting algorithm is prominently used in industry?
  + Quick sort
  + Merge Sort
  + Insertion Sort
  + Bubble Sort