* div\_update(): This function is used to update the color, height and width of the individual array elements (divs) during the visualization procedure. The three main processes- iteration through elements, swapping and update sorted part of array, are colored yellow, red and green respectively. It makes use of this formula to do so:

cont.style= " margin:0% " + margin\_size + "%; width:" + (100/array\_size-(2\*margin\_size)) + "%; height:" + height + "%; background-color:" + color + ";";

* generate\_array(): This function is used to assign random values to the array elements based on the size chosen by the user using the formula:

div\_sizes[i]= Math.floor(Math.random() \* 0.5\*(inp\_as.max - inp\_as.min) ) + 10;

This formula basically scales the randomly generated value to a fixed range and then rounds it off to the lower bound before assigning to the element.

* update\_array\_size(): This function invokes the generate array() function to assign the respective array elements their values. It is invoked by the users pressing of the “generate array” button on the webpage.
* vis\_speed(): This function is used to assign a speed to the visualizer’s animation process using a switch-case with five cases: 1, 2, 3, 4 and 5 which correspond to the input range box in the website and will assign values 1, 10, 200, 1000 and 10000 respectively.
* quick\_partition(): This function is used to implement the divide and conquer algorithm which is an essential part of quick sort. The function is called recursively within the main sorting function until the array is sorted.
* merge\_partition(): This function is used to implement the divide and conquer algorithm which is an essential part of merge sort. The function is called recursively within the main sorting function until the array is sorted.
* enable\_buttons(): This function is used to unlock the sorting algorithms, array size, array speed and the generate array buttons once the execution of the visualizer is completed. If the user presses on these buttons when the program is being executed, the inputs of the different parameters can be altered which will harm the visualization process which is why locking them is necessary. After the execution is completed, the buttons are unlocked for the next execution.
* disable\_buttons(): These functions are used to lock the sorting algorithms, array size, array speed and the generate array buttons during the execution of the visualizer. If the user presses on these buttons when the program is being executed, the inputs of the different parameters can be altered which will harm the visualization process which is why locking them is necessary. After the execution is completed, the buttons will be unlocked for the next execution.
* runalgo(): This function will call the disable\_buttons() function to lock the specified buttons and execute the algorithm that the user selects via switch-case.