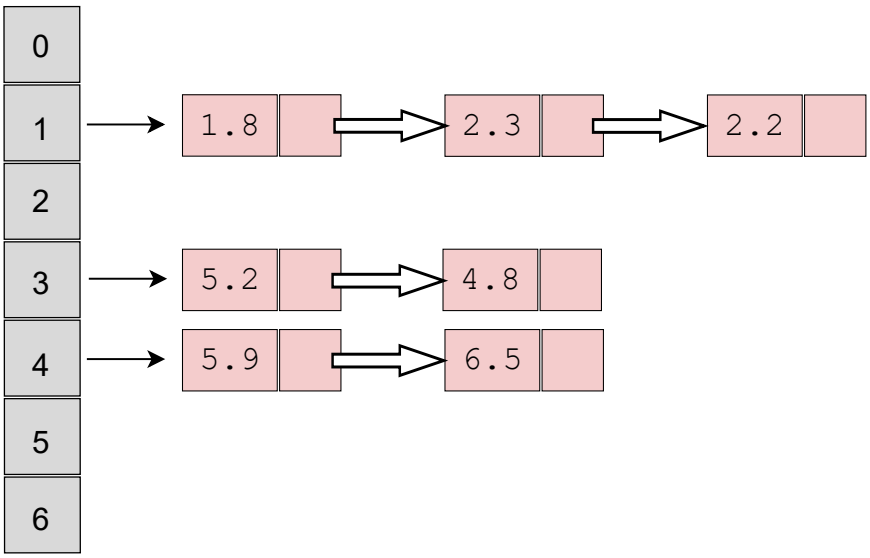
Introduction to Bucket Sort

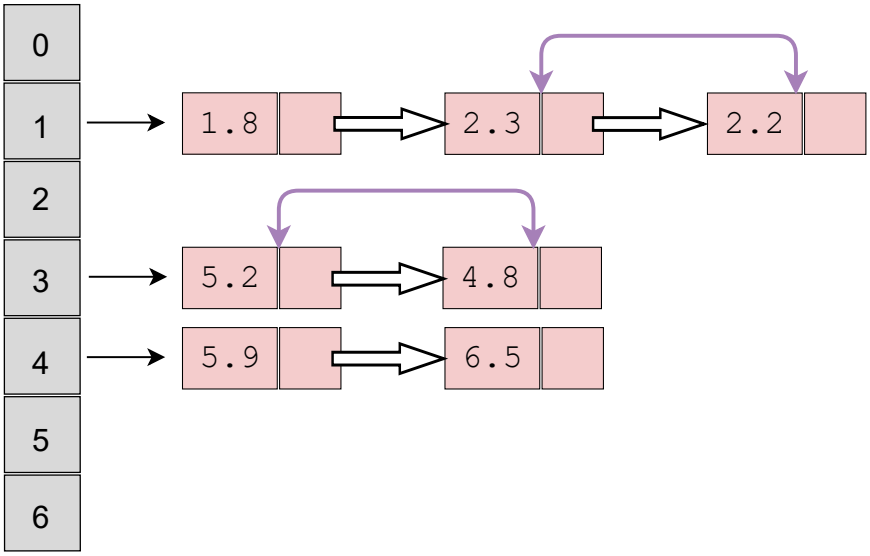
Bucket sort is a very useful sorting algorithm when working with floating point numbers. (Reading time: under 3 minutes)

We store the values in buckets, which we then sort using insertion sort, and merge.

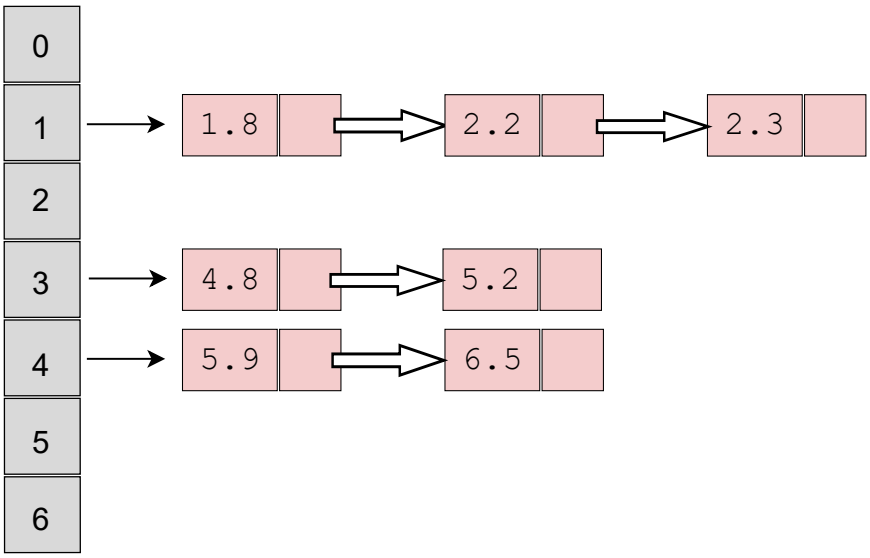
Let's say we want to sort the following array, with floating point numbers as values. First, we create as many buckets as the length of the array, 7 in this case. Then, we store the values based on their value, by (for example) using **Math.floor(n * arr[i] / 10)**. This multiplies the length of the array with the value, and divides it by 10.



Math.floor(n * arr[i] / 10) is 3, so the element with value 5.2 gets stored in bucket 3. This applies to all elements in the array. The items are connected as a linked list.

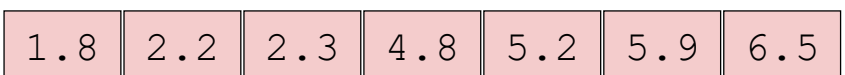


Now, we sort the elements in every individual bucket using insertion sort. Then, all items from the lowest bucket to the highest bucket gets pushed to a new array. This means that the lowest values will be pushed first, and the highest values will be pushed last, which results in a sorted array.



Now, we sort the elements in every individual bucket using insertion sort. Then, all items from the lowest bucket to the highest bucket gets pushed to a new array. This means that the lowest values will be pushed first, and the highest values will be pushed last, which results in a sorted array.

The sorted array becomes:



In the next lesson, I will talk about the implementation of this algorithm.