

## Computer Science and Programming Homework November 30th, 2020

### Task 1 *Sorting Algorithm – Bucket Sort*

*Bucket Sort* is a comparison-type algorithm which assigns elements of a list we want to sort in *Buckets*. The contents of these buckets are then sorted, typically with another algorithm. After sorting, the contents of the buckets are appended, forming a sorted collection. To implement the Bucket Sort, please follow these steps below:

1. Set up a list of empty buckets(=empty list). A bucket is initialized for each element in the input list.
2. Iterate through the input list and insert element to the buckets. Where each element is inserted depends on the input list and the largest element of it. The optimal *size* of each bucket can be obtained by dividing the largest element with the length of the input list. Next, by dividing the element's value with this *size*, we'll get an index for each element's respective bucket.
3. Sort each non-empty bucket. You can do this with any sorting algorithm. Since we're working with a small dataset, each bucket won't have many elements so *Insertion Sort* works well for us here.
4. Visit the buckets in order. Once the contents of each bucket are sorted, after concatenation, they will yield a list in which the elements are arranged as desired.

An example of Bucket Sort can be visualized in Figure 2 and Figure 3.

### Task 2 *k-selection - naive - rst attempt based on sorting*

Implement a function `kselect naive1` to select the  $k$ -th smallest element in an unsorted list which runs in  $O(n^2)$ . The function receives a list  $L$  as input and returns the  $k$ -th smallest element in  $L$ . You can either use sorting or use your own algorithm.

### Task 3 *k-selection - naive - second attempt based on sorting*

Implement a function `kselect naive2` to select the  $k$ -th smallest element in an unsorted list which runs in  $O(n \log n)$ . The function receives a list  $L$  as input and returns the  $k$ -th smallest element in  $L$ . You can either use sorting or use your own algorithm.

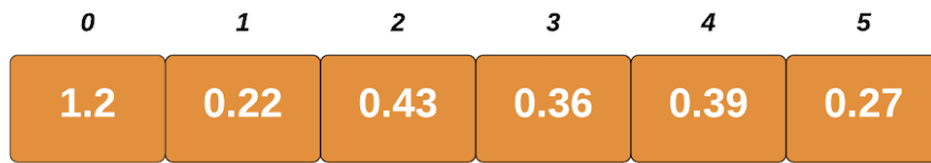


Figure 2: The input list

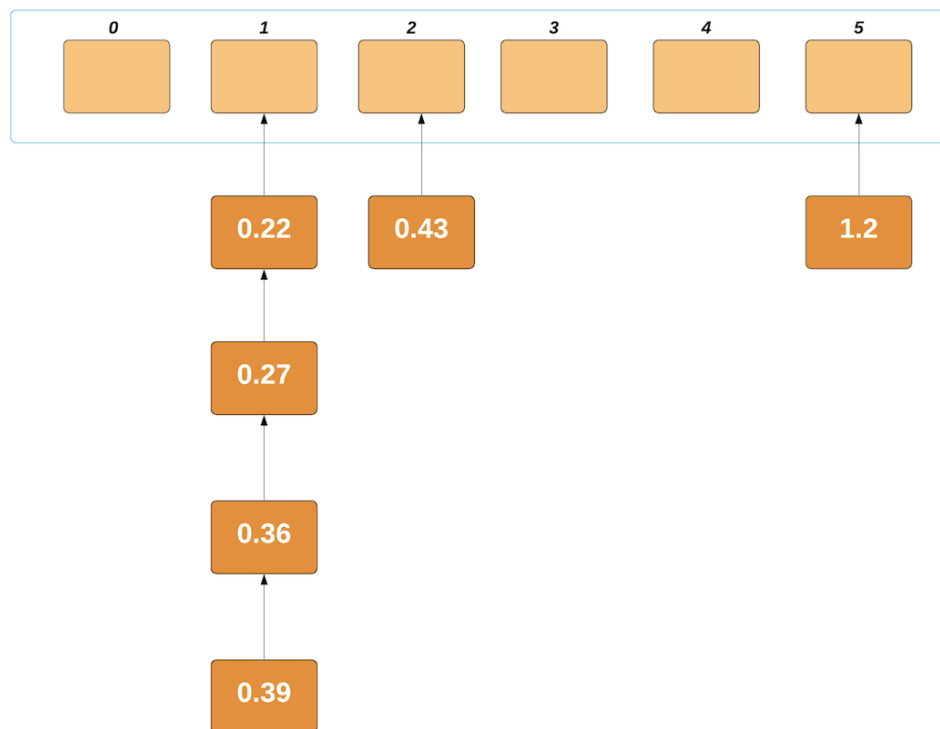


Figure 3: Assignment of elements