Stable algorithms: Insertion sort, merge sort

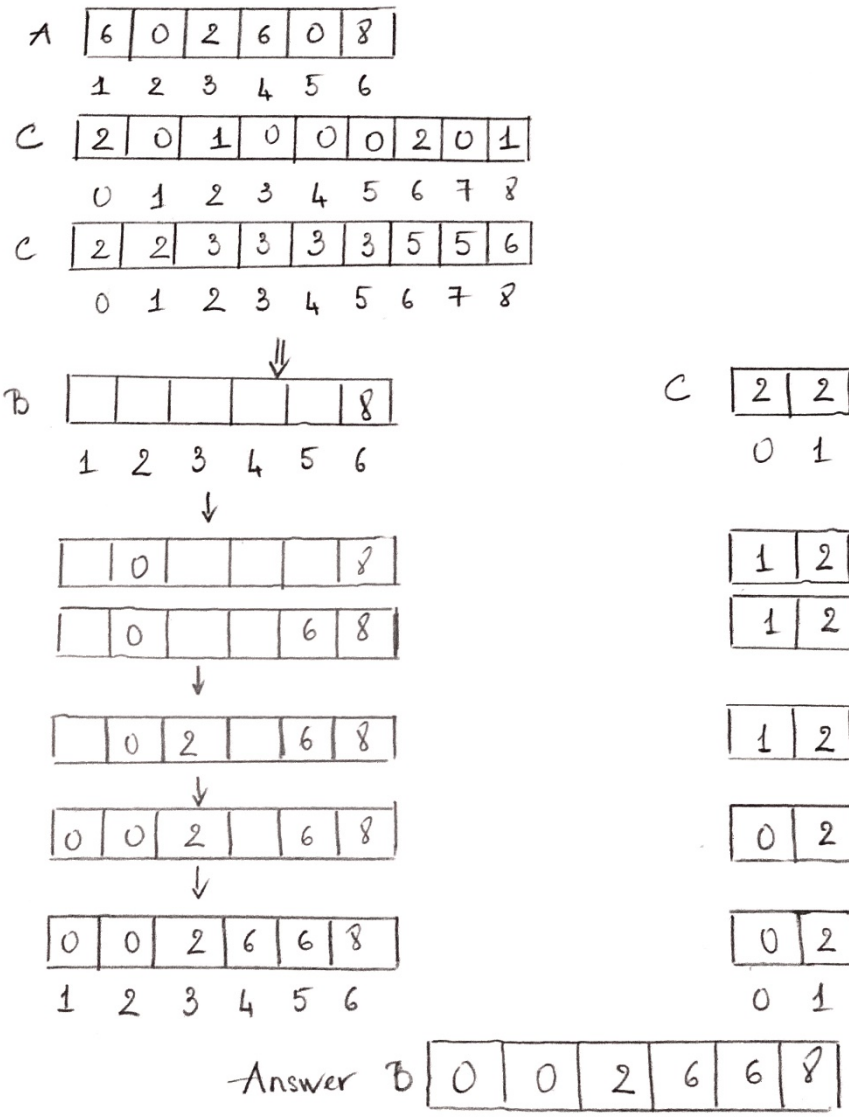
Not stable algorithms: Heap sort, quick sort.

To make algorithm stable, we should store the element with its index (original order) as a secondary way of sorting elements with equal primary value. This scheme takes $O(n \lg n)$ additional space.

2. By Dangnhi Ngo

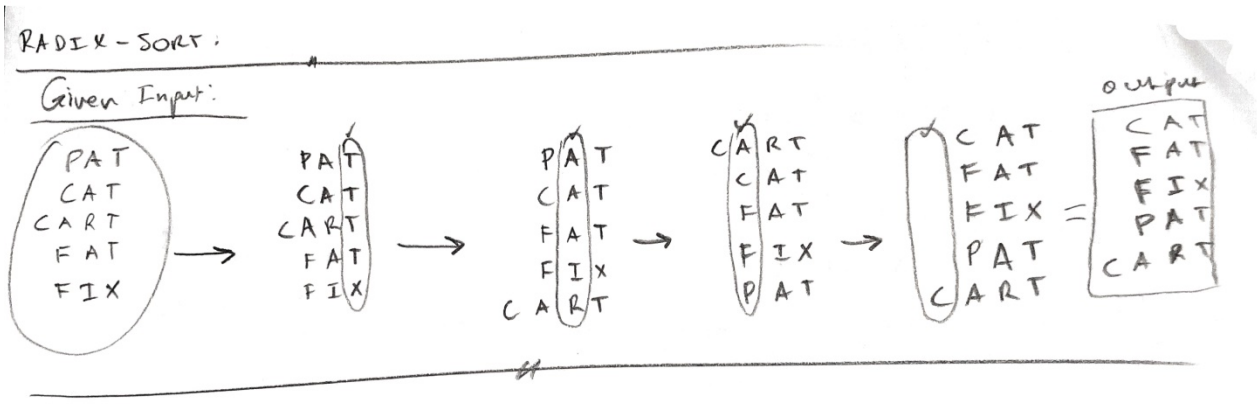
2/ Counting Sort, Radix Sort, Bucket Sort

(1) COUNTING - SORT : $A = \langle 6, 0, 2, 6, 0, 8 \rangle$

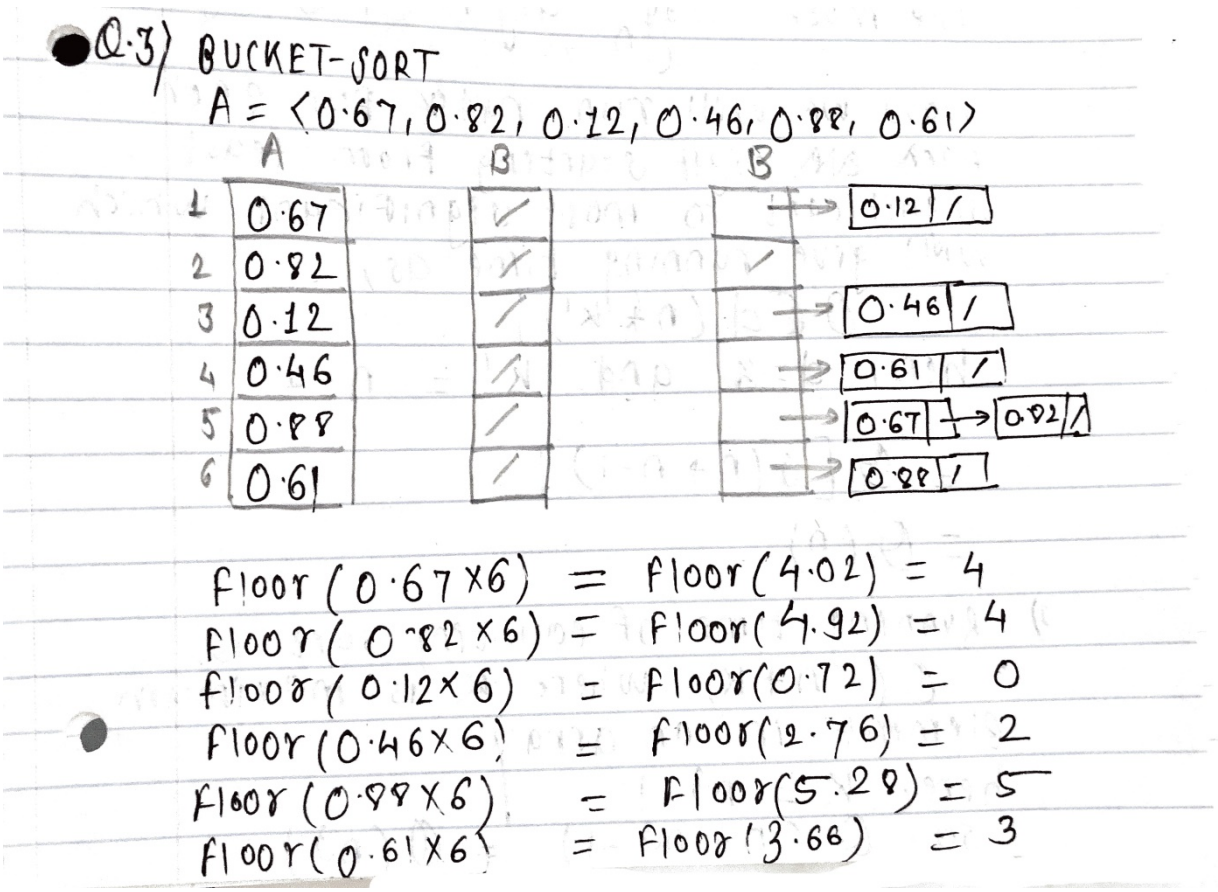


Answer B | 0 | 0 | 2 | 6 | 6 | 8 |

By Vlad Gordiyevsky



By Shraddha Kharche



3. By Duyen Tran

(1)

Using radix sort we have 3 digits in base n so we call counting sort three times

4.

$O(n^3)$ because the range of the input is $n^3 - 1$

$O(n+K)$ K = range of input

$= O(n + (n^3 - 1))$

$= O(n^3)$

4. **Sorting** – Explain why the worst-case running time for bucket sort is $\theta(n^2)$. What simple change to the algorithm preserves its linear average-case running time and makes its worst-case running time $O(n \lg n)$?

   The worst-case running time for bucket sort occurs when a single bucket contains all n elements of the original array. After placing the elements into their appropriate bucket, Insertion-Sort is called to sort them in the bucket which has a worst-case running time of $O(n^2)$. The dominating cost of Bucket-Sort is in sorting each bucket so that can be easily fixed by replacing Insertion-Sort with a different sorting algorithm that has a better worst-case running time. For instance, merge sort has a worst-case running time of $O(n \lg n)$ and can be called to sort each bucket to give Bucket-Sort a worst-case running time also of $O(n \lg n)$.