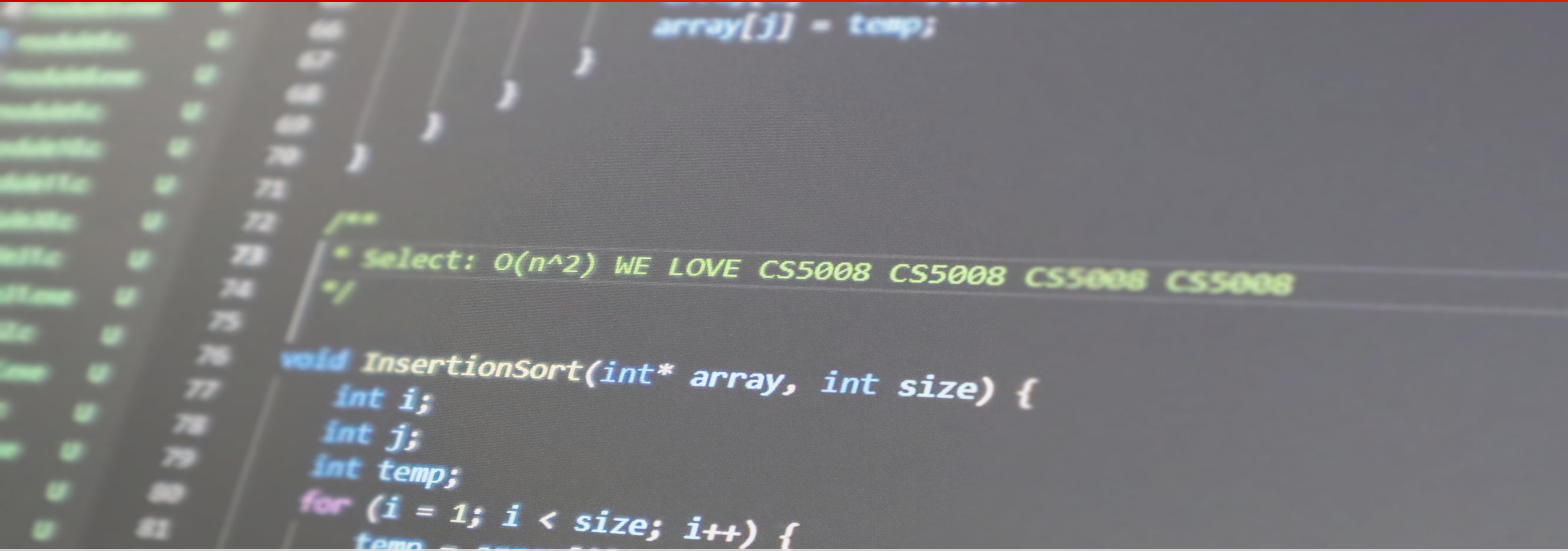
Northeastern University



Conceptual Presentation: Bucket Sort

Cuiting Huang, Zhe Cheng, Hongyang Ye

Review

	Average Time Complexity	Best Case	Worst Case	Space Complexity
Insertion sort	$O(n^2)$	$O(n)$	$O(n^2)$	$O(1)$
Bubble sort	$O(n^2)$	$O(n)$	$O(n^2)$	$O(1)$
Selection sort	$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(1)$
Merge sort	$O(nlog(n))$	$O(nlog(n))$	$O(nlog(n))$	$O(n)$

Comparison sorts

- **Shared Property**: The sorted order they determine is based only on comparisons between the input elements

$$[3, 1, 2] \xrightarrow{(3 < 1)?} [1, 3, 2] \xrightarrow{(3 < 2)?} [1, 2, 3] \xrightarrow{(1 < 2)?} [1, 2, 3]$$

RUNOOB.COM Sorting Algorithm https://www.runoob.com/w3cnote/ten-sorting-algorithm.html
CLRS Chapter 8, Sorting in Linear Time, page 191

- Can we sort even faster? (In linear time $O(n)$)
- Can we sort without comparing two elements?

Bucket Sort

- When we try to sort, we have to **iterate the data set.** Otherwise, we can't sort it properly.
- It means that the fastest time complexity when we sort is **a linear time** $O(n)$.
- So is there any sorting algorithm whose time complexity in Big-Oh is $O(n)$?

Bucket Sort

Bucket Sort

- For bucket sort, instead of comparing, we do something else with the elements in the data set.
- Notice: to guarantee the O(n) run time

Assumption

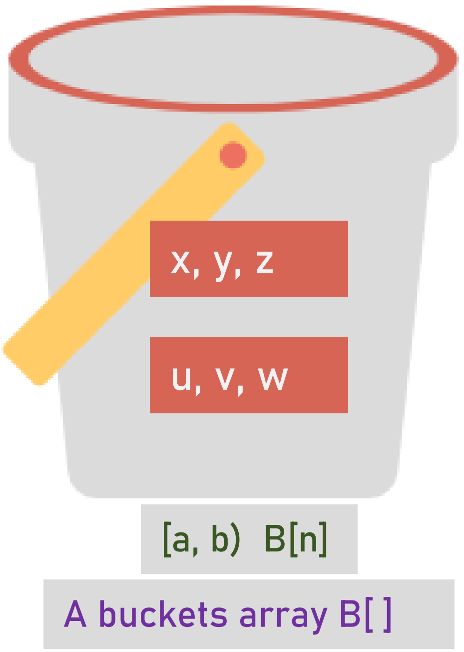
- The data set is generated by a random process, which distributes elements independently over the interval.
- Only the elements are as evenly distributed as possible, the time complexity of bucket sort would be O(n).

What’s a ‘Bucket’?

Bucket Sort-Basic idea

- Bucket is a container with a certain volume, each bucket has a capacity. We could also say that **each bucket represents an interval**. So there might be one or multiple elements in each bucket.

- **First step** -creates some buckets and determine the range of each bucket:
- **How to set up the right number of buckets?**
  - There are **many different ways** to determine the number of buckets.
  - One rule we can take into account is that **it is best to have elements evenly distributed in each bucket.**
  - In the case below, we set up buckets using **Simple Bucket Splitting**



Wikepedia bucket sort https://en.wikipedia.org/wiki/Bucket_sort
NIST's Dictionary of Algorithms and Data Structures: bucket sort https://xlinux.nist.gov/dads/HTML/bucketsort.html

Bucket Sort-Basic idea

- **First step-**create Buckets *Simple Bucket Splitting
- Find maximum and minimum value, max=4.5 min=0.5 dif=4.0
- Set the case of bucket splitting as **bucketNum=max/k-min/k +1 (cause max<10, k=1)**

  round down to an integer bucketNum=5
- Calculate the range of each bucket. **Interval span =(man-min)/(bucketNum-1)= 4 / 4 = 1**

A Double Data Set :4.5   0.84   3.25   2.18   0.5

A buckets array B[ ]

[0.5, 1.5) B[0]

[1.5, 2.5) B[1]

[2.5, 3.5) B[2]

[3.5, 4.5) B[3]

[4.5, 5.5] B[4]

Geeks for Geeks https://www.geeksforgeeks.org/bucket-sort-2/
Bucket Sort Algorithm https://www.youtube.com/watch?v=7mahJ1axrR8

Bucket Sort-Basic idea

- **Second step-**traverse the data set and place the elements into the corresponding bucket.
- Here we will use a mapping function, and each element will be mapped to the n^{th} bucket.
- **the subscript n = (array[i] - min) * (bucketNumber-1) / (max-min)**

(4.50-0.5)*4/4=4
(0.84-0.5)*4/4=0
(3.25-0.5)*4/4=2
(2.18-0.5)*4/4=1
(0.50-0.5)*4/4=0

A Double Data Set :4.5 0.84 3.25 2.18 0.5

A buckets array B[ ]

[0.5, 1.5) B[0]	[1.5, 2.5) B[1]	[2.5, 3.5) B[2]	[3.5, 4.5) B[3]	[4.5, 5.5] B[4]

Geeks for Geeks https://www.geeksforgeeks.org/bucket-sort-2/
Bucket Sort Algorithm https://www.youtube.com/watch?v=7mahJ1axrR8

Bucket Sort-Basic idea

- **Second step-**distribute elements into corresponding buckets.

- In these steps what we need to focus on is the mapping function, it is the key to making our algorithm more efficient.

- In this case,

bucketNum =max/k-min/k +1 (cause max<10, k=1)

f(c)=(array[i] - min) * (bucketNumber-1) / (max-min) = (int) (array[i]-min)* 1/k

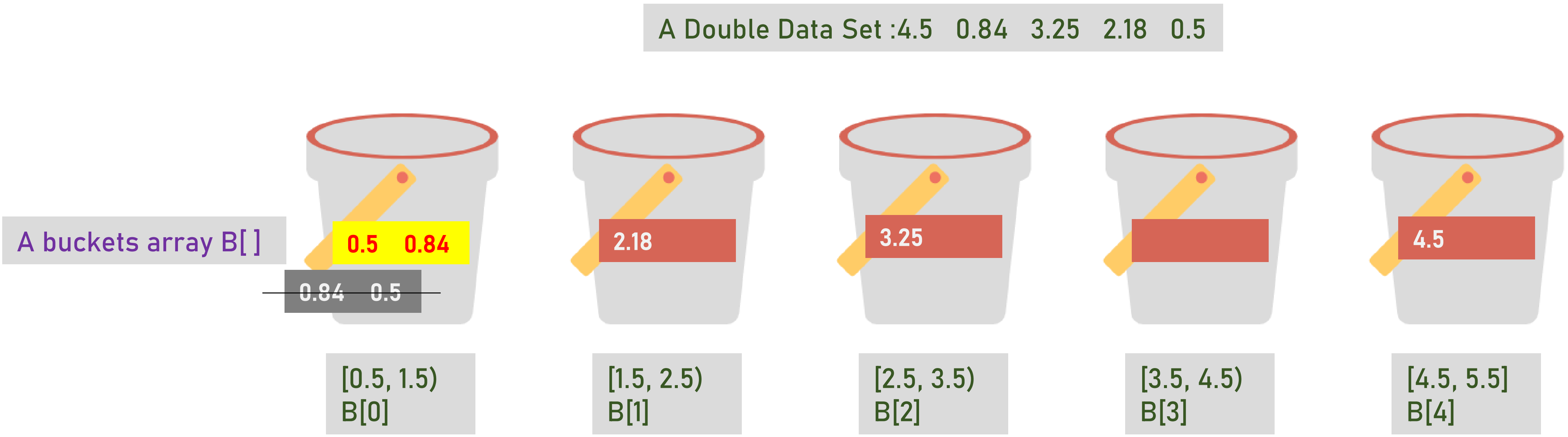
- It is obvious that the determination of the mapping function has a great relationship with **the characteristics of the data set.**

Geeks for Geeks https://www.geeksforgeeks.org/bucket-sort-2/
Bucket Sort Algorithm https://www.youtube.com/watch?v=7mahJ1axrR8

Bucket Sort-Basic idea

- **Third Step-**sort the elements inside each bucket separately.

- Here you might have a question:
  - With using comparing sort algorithm, does it have an impact on the time complexity of bucket sort?

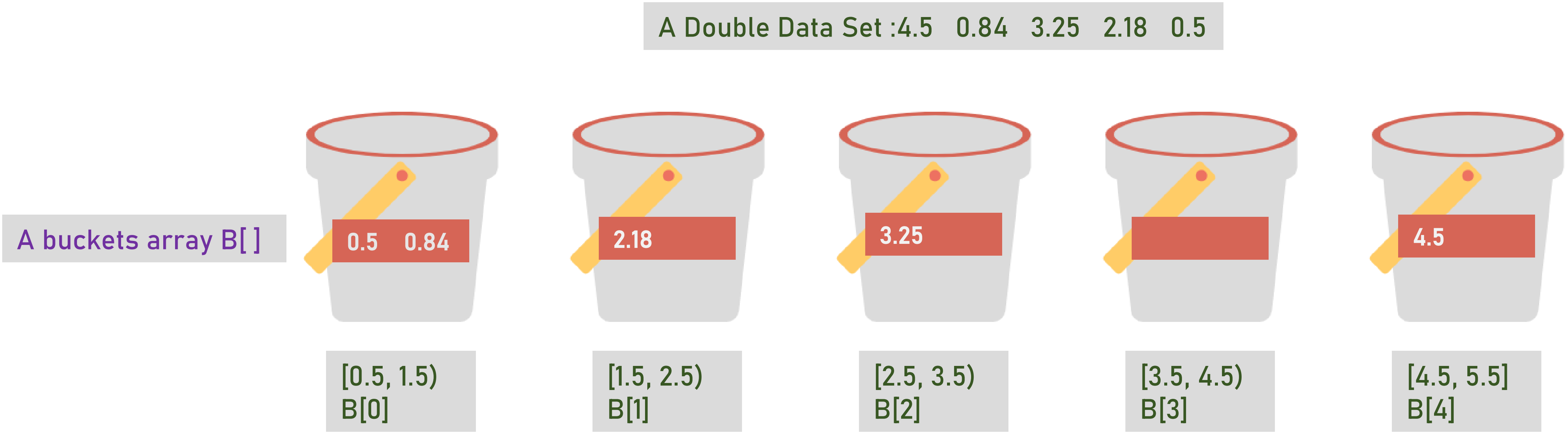


Geeks for Geeks https://www.geeksforgeeks.org/bucket-sort-2/
Bucket Sort Algorithm https://www.youtube.com/watch?v=7mahJ1axrR8

Bucket Sort-Basic idea

- **Last Step-**iterate over all the buckets and put all elements back to the data set.

- we get the data set = { 0.5, 0.84, 2.18, 3.25, 4.5}



Geeks for Geeks https://www.geeksforgeeks.org/bucket-sort-2/
Bucket Sort Algorithm https://www.youtube.com/watch?v=7mahJ1axrR8

Bucket Sort-Basic idea

- **First step-**create these buckets.

  bucketNum =(int) (max/k-min/k +1), f(n)=(int) (array[i]-min)* 1/k

- **Second step-**distribute elements into corresponding buckets.
- **Third Step-**sort the elements inside each bucket separately.
- **Last Step-**iterate over all buckets and print all elements in turn.

- **Notice that:**
  - bucket sort is an out-of-place algorithm, the number of buckets should be just enough, neither overflowing nor too little.
  - another way to set up buckets is **Reduced bucket splitting.**

Geeks for Geeks https://www.geeksforgeeks.org/bucket-sort-2/
Bucket Sort Algorithm https://www.youtube.com/watch?v=7mahJ1axrR8

Bucket Sort-Reduced buckets splitting

- If the range of the data set is relatively large, such as input [103, 9, 105, 1, 7, 101, 205, 201, 209, 107, 5] (size:11)
  - Using simple buckets splitting, bucketNum=209/10 -1/10 = **21**, f(n)=(value-min)/10
  - The bucketNum is extremely **overflowing**, there will be a bunch of empty buckets in the middle.

- In this case, we should set up the Reduced buckets:

max=209, min=1, dif=208, size = 11;

the mapping function be like f(n) =(array[i] -min)/(max-min) * array.length

Dynamic description for Bucket Sort

https://youtu.be/vt1YX_ndHMk

Cuiting Huang

Bucket Sort-data sets

- **What kind of data could we use bucket sort?**
- In addition to the numerical array, bucket sort can be used dealing with a **String array**

- For example, there is a string array S[D, a, F, 99, B, c, A, z],
  - All lowercase letters are required to precede uppercase letters, but no order is required between lowercase letters and uppercase letters. Then put the digits behind the letters.

  - By using bucket sort, we can separate letters and digits

Northeastern University

Dynamic description for Bucket Sort

the type of data set： String

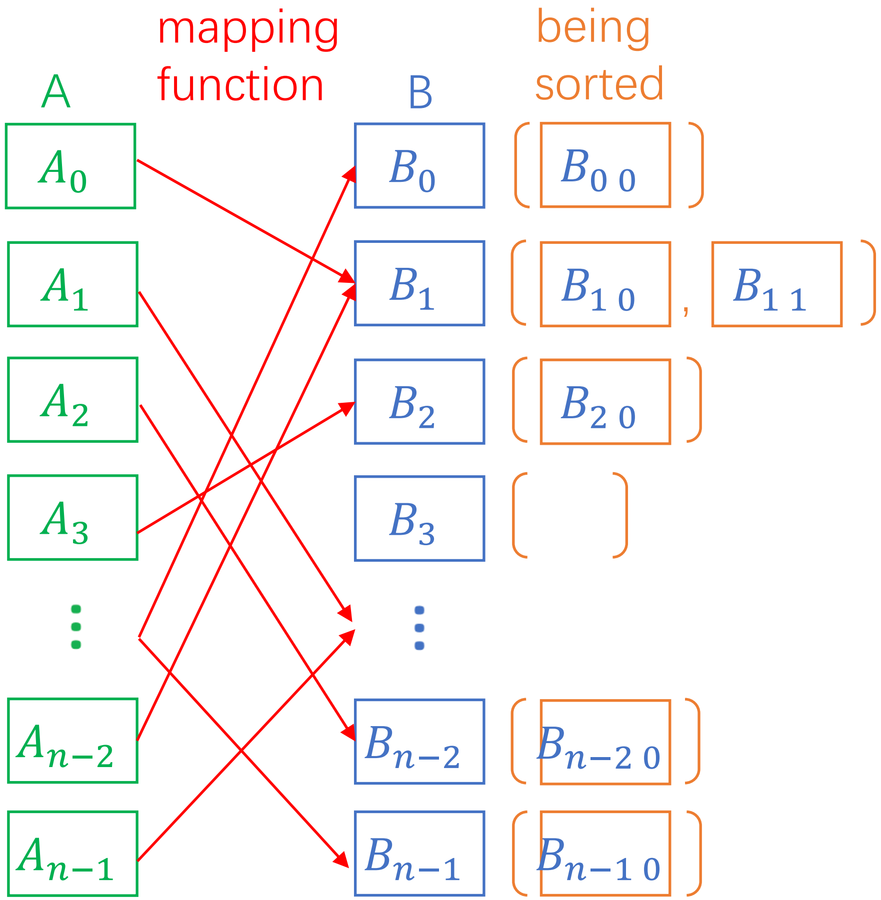
https://youtu.be/Ggf4NUe7bCg

Cuiting Huang

Bucket Sort – How to implement one

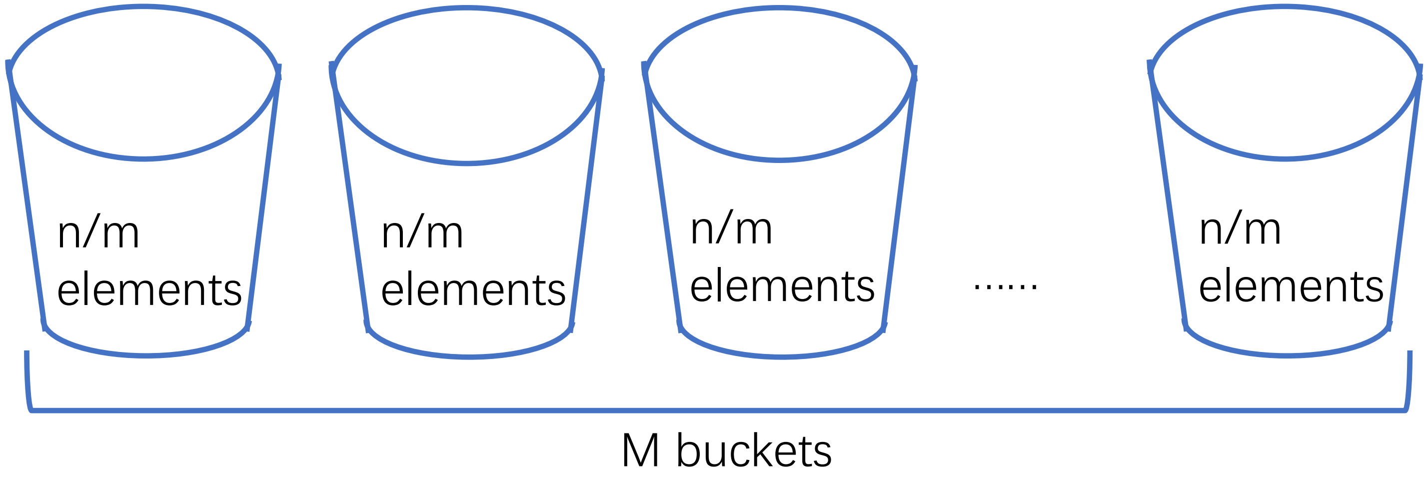
Bucket – sort(A)

1. *let* $B[0 \ldots n1]$ *be a new array*
2. $n = A.length$
3. *for i = 0 to n – 1*
4. *make* $B[i]$ *an empty list*
5. *for i = 0 to n – 1*
6. *insert* $A[i]$ *into list* $B\left[nA[i]\right]$
7. *for i = 0 to n – 1*
8. *Sort list* $B[i]$ *with insertion sort*
9. *Concatenate the lists* $B[0], B[1], \ldots, B[n - 1]$ *together in order*



Bucket Sort – Practices

- There will be n/m elements in each bucket. Nonetheless, programmers often utilize length of input array as the number of buckets.
- To avoid comparisons, we should find ways to have up to 1 element in 1 bucket.



Bigsai, Understanding bucket sort in three minutes. https://www.cnblogs.com/bigsai/p/13396391.html

Bucket Sort – Time complexity

In an average case, the time complexity is like as such:

$$O(N) + \{\ \boxed{O(M * (N/M) * \log(N/M))}\ \}$$

$$= O(N) + \{\ \boxed{O(N * \log(N/M))}\ \}$$

If comparison avoidance is done perfectly, the time complexity is:

$$O(n)$$

- ✓ So be it, if the sort algorithm applied upon elements inside a bucket has average-case time complexity: $O(nlogn)$
- ✓ To a very extent, the overall time complexity is determined by this chunk!

Bucket Sort – Space complexity

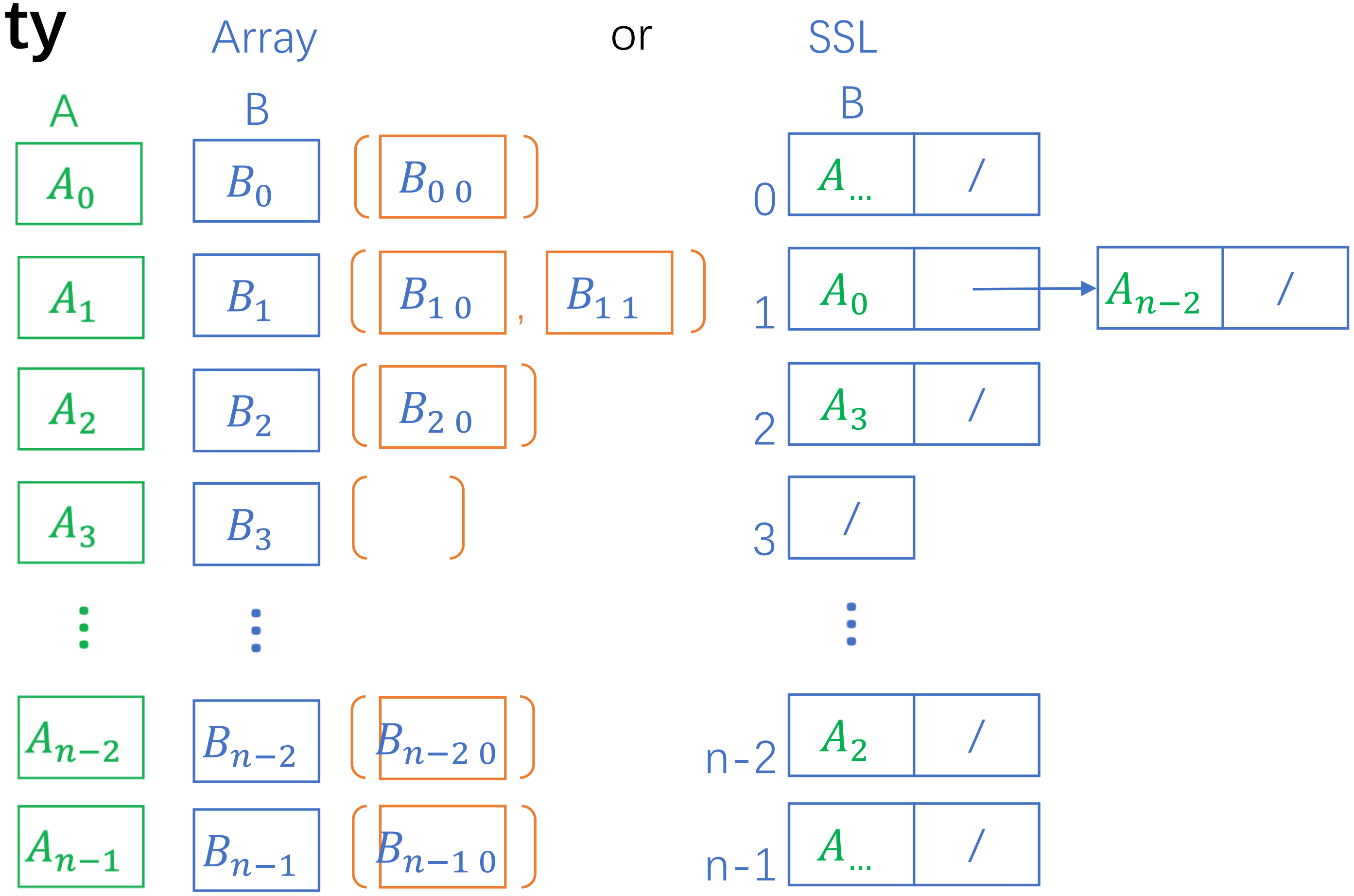
Space complexity really depends on how the buckets are presented.

I. array
II. Singly linked list

In worst cases, we will have O(N*M) space complexity.

Space complexity increases when

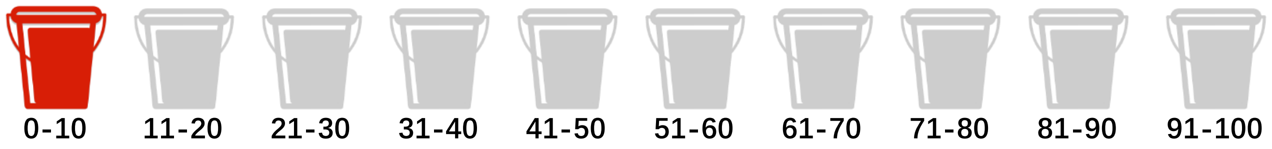
I. input array elements increases
II. Number of buckets increases

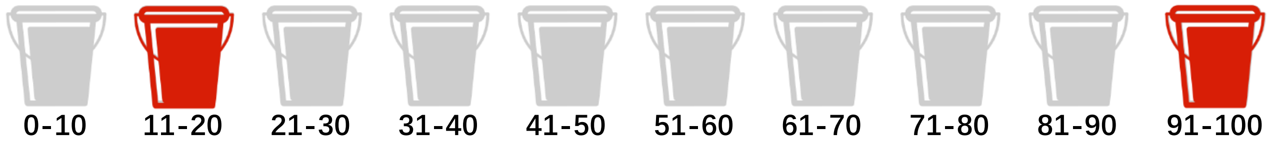


Bucket Sort – Pros and Cons

Pros:

- Can sort in linear time if the data set satisfies the assumption:
  - The data set conforms to uniform distribution





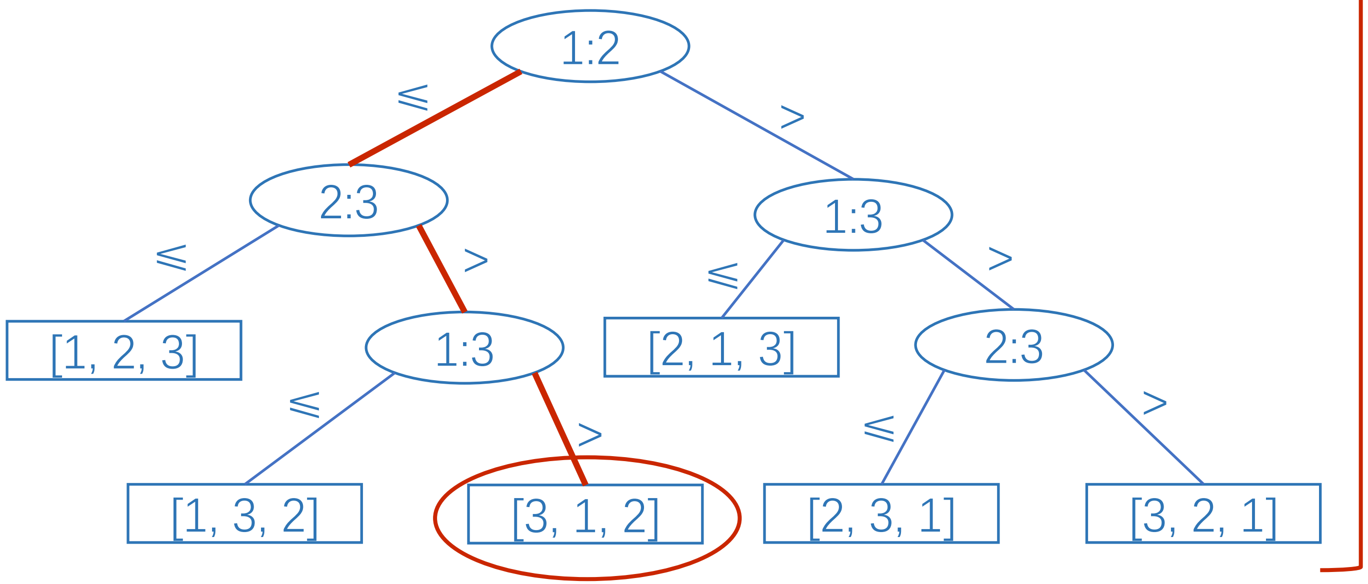
Cons:

- Can perform badly if all the inputs fall into one bucket.
  - [0, 1, 1, 1, 1, 1, 1, 1, 1, 1]
- Extra space for buckets is needed
  - [11, 19, 15, 17, 13, 12, 94, 96, 98, 99]

- What's the difference between linear sort and comparison sort?
- Why bucket sort can beat the lower bound of comparison sort?

The lower bound of comparison sort

- Example: [6, 8, 5]

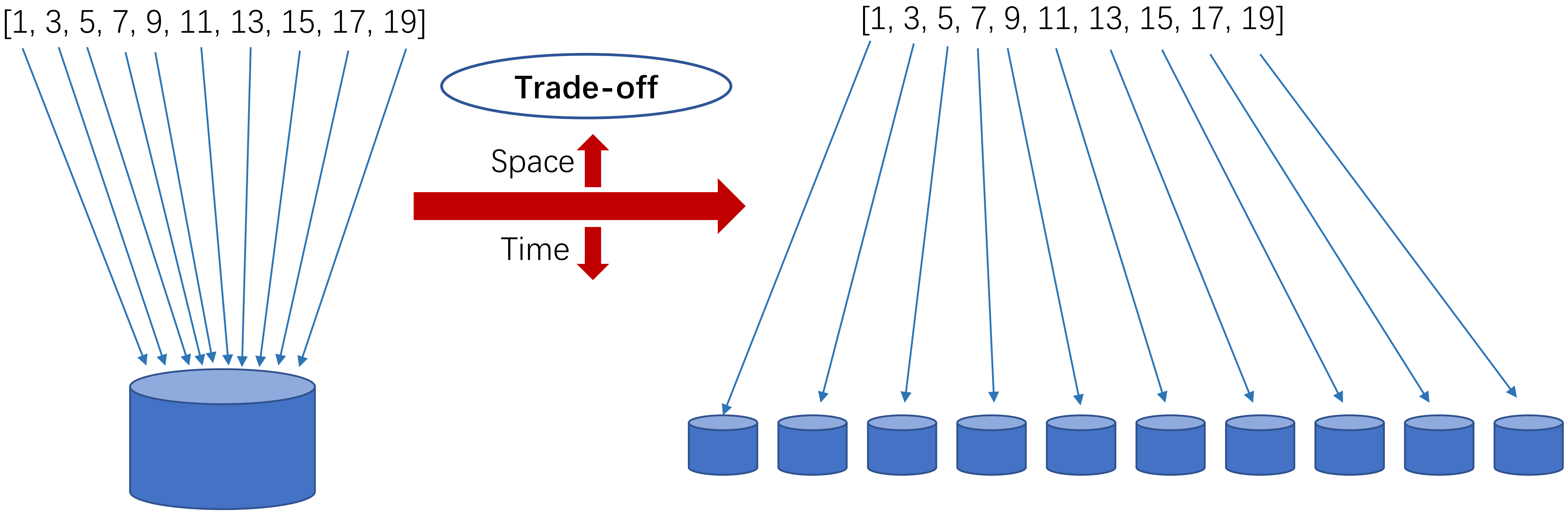


The worst-case number of comparisons for a given comparison sort algorithm equals the height of its decision tree $\left(nlog(n)\right)$.

- **Theorem**: Any comparison sort algorithm requires $\Omega(nlg(n))$ comparisons in the worst case.

Linear sort – Beat the lower bound

- We do not compare!



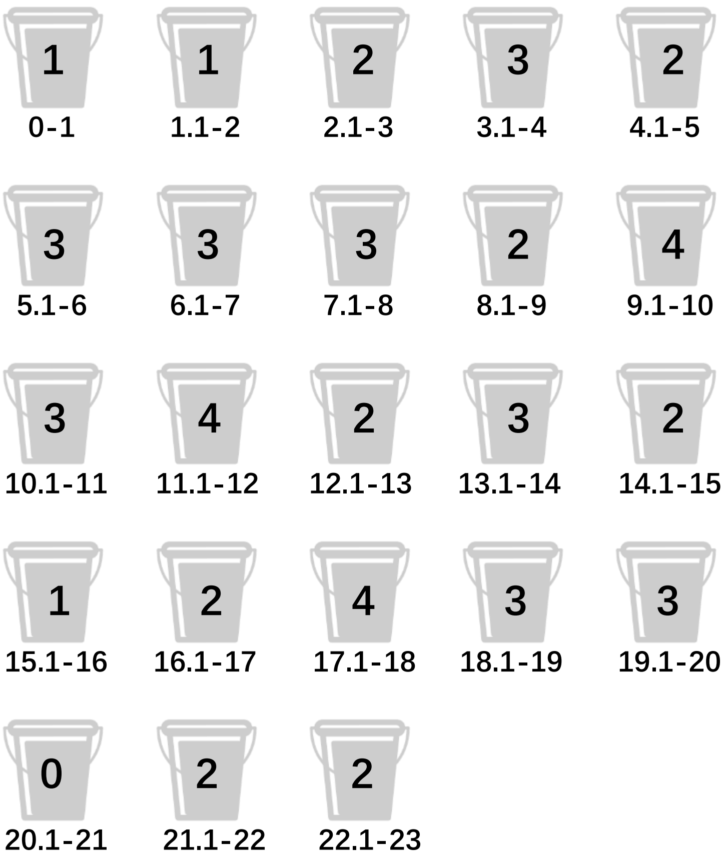
Real-world application

When to use bucket sort?

- The data set has some specific features.

Example:

- The data in the table below are 55 smiling times, in seconds, of an eight-week-old baby.
- The smiling times, in seconds, follow a uniform distribution between zero and 23 seconds, inclusive.



10.4	19.6	18.8	13.9	17.8	16.8	21.6	17.9	12.5	11.1	4.9
12.8	14.8	22.8	20.0	15.9	16.3	13.4	17.1	14.5	19.0	22.8
1.3	0.7	8.9	11.9	10.9	7.3	5.9	3.7	17.9	19.2	9.8
5.8	6.9	2.6	5.8	21.7	11.8	3.4	2.1	4.5	6.3	10.7
8.9	9.4	9.4	7.6	10.0	3.3	6.7	7.8	11.6	13.8	18.6

Lumenlearning, Introduction to statistics, the uniform distribution, https://courses.lumenlearning.com/odessa-introstats1-1/chapter/the-uniform-distribution/

Real-world application

When to use comparison sort?

- The data set does not satisfy the assumptions of linear sort
- We do not have much space for sorting.

Example:

Sorting 3 million domain names based on theirs lengths.

- We do not use linear sort to avoid memory leak.
- We might use merge sort or quick sort to make the sorting process as fast as possible.

Conclusion

- Sorting algorithm comparing two elements can be categorized as **comparison sort**.
- Linear sort can be faster than comparison sort, with $\boldsymbol{O(n)}$**.**
- The critical assumption of bucket sort is:
    - The data is as evenly distributed as possible
- **Designing the mapping function** of bucket sort is of vital importance
- The lower bound of comparison sort is $\boldsymbol{O(nlog(n))}$**.** Linear sort can beat this bound because **it does not compare**.
- The limits of bucket sort are:
    - Can perform badly if all the inputs fall into one bucket.
    - Extra space for buckets is needed

Q & A

References

[1] T. Cormen, C. Leiserson, R. Rivest, C. Stein. *“Sorting in Linear Time” in Introduction to Algorithms,* 3rd ed. Cambridge, Massachusetts & London, England: The MIT Press, 2009. pp. 191 – 212.

[2] Hustcc, “Ten classic sorting algorithms.” GitHub.https://github.com/hustcc/JS-Sorting-Algorithm(accessed Nov. 5, 2021).

[3] Bigsai, “Understanding bucket sort in three minutes.”Cnblogs.https://www.cnblogs.com/bigsai/p/13396391.html(accessed Nov. 5, 2021).

[4] Wikepedia, “Bucket sort.”Wikipedia.https://en.wikipedia.org/wiki/Bucket_sort(accessed Nov. 5, 2021).

[5] Paul E. Black, "bucket sort in Dictionary of Algorithms and Data Structures.”Nist. https://www.nist.gov/dads/HTML/bucketsort.html(accessed Nov. 5, 2021).

[6] Geeks for Geeks, “Bucket sort.”GeeksforGeeks.https://www.geeksforgeeks.org/bucket-sort-2/(accessed Nov. 6, 2021).

[7] Simplilearn, “Bucket Sort Algorithm.”Youtube.https://www.youtube.com/watch?v=7mahJ1axrR8(accessed Nov. 6, 2021).

References

[8] Ladinirenbeliz, “What is bucket sort?”Educative.https://www.educative.io/edpresso/what-is-bucket-sort(accessed Nov. 6, 2021).

[9] Quora, “How can I determine how many bucket I need for a bucket sort?”Quora.https://www.quora.com/How-can-I-determine-how-many-buckets-I-need-for-a-bucket-sort(accessed Nov. 15, 2021).

[10] ICS 161, “Bucket Sorting.”Isc.https://www.ics.uci.edu/~eppstein/161/960123.html(accessed Nov. 15, 2021).

[11] Lumenlearning, “Introduction to statistics, the uniform distribution.”LumenLearning.https://courses.lumenlearning.com/odessa-introstats1-1/chapter/the-uniform-distribution/(accessed Nov. 6, 2021).

[12] Liuxinyu, “Introduction to bucket sort.”Cnblogs.https://www.cnblogs.com/liuxinyustu/articles/13563211.html(accessed Nov. 6, 2021).

[13] Javarevisited, “Difference between comparison and non-comparison based sorting algorithms? Example”Javarevisited.https://javarevisited.blogspot.com/2017/02/difference-between-comparison-quicksort-and-non-comparison-counting-sort-algorithms.html#axzz7BtbYZHXw(accessed Nov. 6, 2021).

[14] William Zhu, “Application of sorting algorithm.”Csdn.https://blog.csdn.net/allwefantasy/article/details/3062270(accessed Nov. 6, 2021).