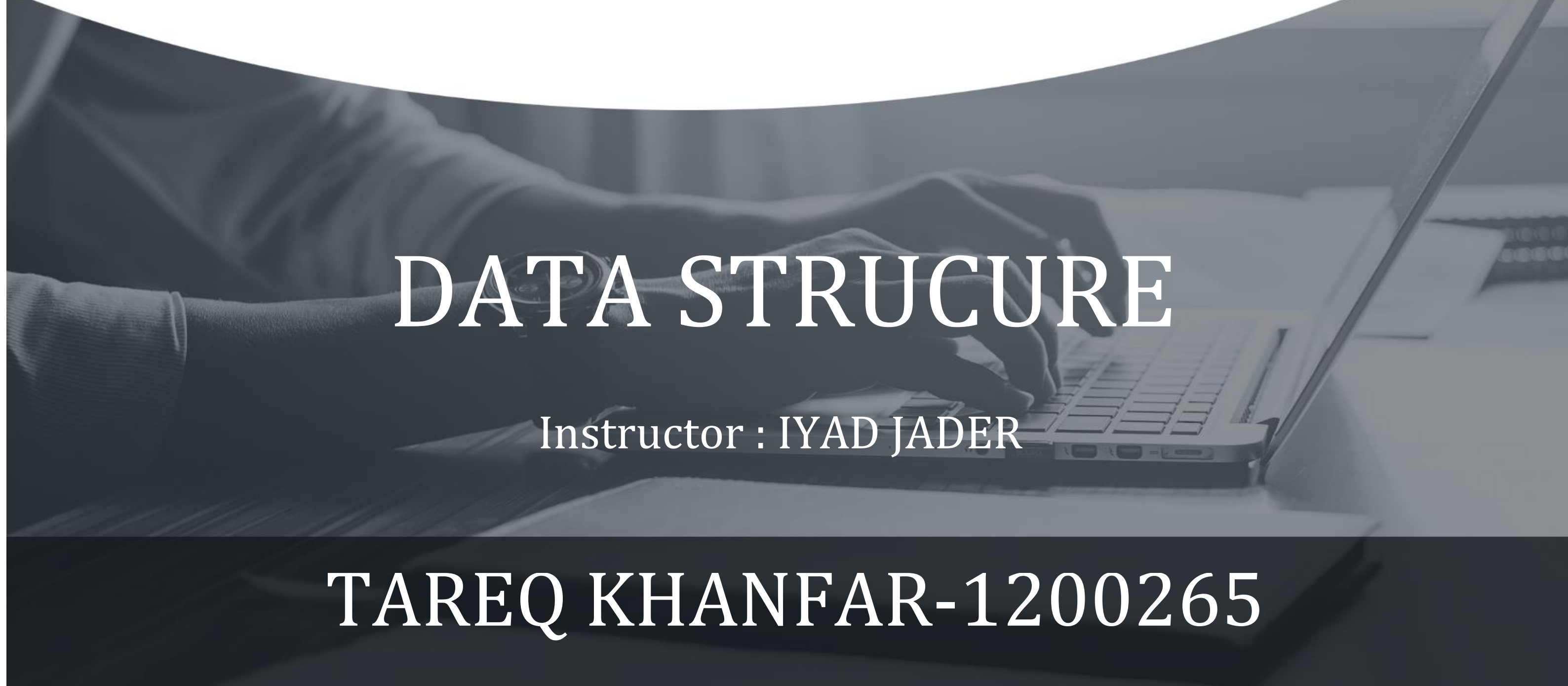
SORTING ALGORITHIM

2022



DATA STRUCURE

Instructor : IYAD JADER

TAREQ KHANFAR-1200265

Table of Contents

Patience Sort

Patience sorting is a sorting algorithm based on card game Patience. In this sorting algorithm the rules of patience game is used to sort an list of elements based on their values.(1)

Rules of Patience Game:

- Cards with lower value can be placed over the card.
- If there is no possible position for a card, then a new pile can be created.
- Goal is to form as much as few piles possible.

Patience Sorting: There are generally two steps in the patience sorting that is creation of the piles and merging the piles.

Algorithm :

Initialize a 2D array to store the piles.

- Traverse the given array and perform the following operations:
    1. Iterate over all the piles and check the top of the stack of each pile is less than the current element or not. IF found to be true, then push the current element to the top of the stack.
    2. Otherwise, create a new pile with the current element as the top of that stack .

Analysis (2)

Data Structure	Time Best Case	Time Avg Case	Time Worst Case	Space complexity	Stability
Array	O(n)	O(n log n)	O(n log n)	O(n)	Yes

Comb Sort :

Comb Sort is mainly an improvement over Bubble Sort. Bubble sort always compares adjacent values. So all inversions are removed one by one. Comb Sort improves on Bubble Sort by using gap of size more than 1. The gap starts with a large value and shrinks by a factor of 1.3 in every iteration until it reaches the value 1

The basic idea is to eliminate turtles, or small values near the end of the list, since in a bubble sort these slow the sorting down tremendously. Rabbits, large values around the beginning of the list, do not pose a problem in bubble sort.
i.e. comb sort is that the gap can be much more than 1. The inner loop of bubble sort, which does the actual swap, is modified such that the gap between swapped elements goes down (for each iteration of outer loop) in steps of a "shrink factor" k: [n/k, n/k2, n/k3, ..., 1] . (3)

Analysis : (4)

Data Structure	Time Best Case	Time Avg Case	Time Worst Case	Space complexity	Stability
Array	O(n log n)	O (n^2 / 2 ^p)	O(n^2)	O(1)	Yes

Counting Sort

Counting sort is a sorting algorithm that sorts the elements of an array by counting the number of occurrences of each unique element in the array. The count is stored in an auxiliary array and the sorting is done by mapping the count as an index of the auxiliary array.

Counting sort is effective when range is not greater than number of objects to be sorted. It can be used to sort the negative input values.
Counting sort makes assumptions about the data, for example, it assumes that values are going to be in the range of 0 to 10 or 10 – 99 etc, Some other assumptions counting sort makes are input data will be all real numbers.
Like other algorithms this sorting algorithm is not a comparison-based algorithm, it hashes the value in a temporary count array and uses them for sorting.
It uses a temporary array making it a non In Place algorithm. (5)

Algorithm : (6)

1 -Find out the maximum element (let it be `max`) from the given array
2 -Initialize an array of length `max+1` with all elements 0. This array is used for storing the count of the elements in the array.

3 -Store the count of each element at their respective index in `count` array

For example: if the count of element 3 is 2 then, 2 is stored in the 3rd position of `count` array. If element "5" is not present in the array, then 0 is stored in 5th position.

4 -Store cumulative sum of the elements of the count array. It helps in placing the elements into the correct index of the sorted array.

5 -Find the index of each element of the original array in the count array. This gives the cumulative count. Place the element at the index calculated as shown in figure below

6 - After placing each element at its correct position, decrease its count by one. in count array.

Analysis : (6)

Data Structure	Time Best Case	Time Avg Case	Time Worst Case	Space complexity	Stability
Array	O(n + k)	O(n + k)	O(n + k)	O(MaxValue)	Yes

Pigeonhole Sort

Pigeonhole sorting is a sorting algorithm that is suitable for sorting lists of elements where the number of elements and the number of possible key values are approximately the same.
It requires O(*n* + *Range*) time where n is number of elements in input array and 'Range' is number of possible values in array.(7)

Algorithim : (7)

1. Find minimum and maximum values in array. Let the minimum and maximum values be 'min' and 'max' respectively. Also find range as 'max-min+1'.

2. Set up an array of initially empty "pigeonholes" the same size as of the range.
3. Visit each element of the array and then put each element in its pigeonhole. An element arr[i] is put in hole at index arr[i] – min.
4. Start the loop all over the pigeonhole array in order and put the elements from non- empty holes back into the original array.

Analysis : (8)

Data Structure	Time complexity	Space complexity	Stability
Array	O(n + Range)	O(n + Range)	Yes

Stability : Yes is suitable for sorting lists of elements where the number of elements (*n*) and the length of the range of possible key values (*N*) are approximately the same

Bucket Sort

In the Bucket Sorting technique, the data items are distributed in a set of buckets. Each bucket can hold a similar type of data. After distributing, each bucket is sorted using another sorting algorithm. After that, all elements are gathered on the main list to get the sorted form.(9)

Algorithim : (9)

1-Find maximum element and minimum of the array
2-Calculate the range of each bucket
3-Create n buckets of calculated range
4. Scatter the array elements to these buckets
5. Now sort each bucket individually
6. Gather the sorted elements from buckets to original array

Analysis : (10)

Data Structure	Time Best Case	Time Avg Case	Time Worst Case	Space complexity	Stability
Array	O(n + k)	O(n)	O(n ^2)	O(n + k)	Yes

Summary :

comparison between comb sort and Bucket sort

They both have the same Time complexity O(n^2) and the difference between them

Space Complexity in comb sort is O(1) because dependent in comparison

between elements . and Bucket sort the space Complexity is O(N+K) because

Bucket sort is a non-comparison based sorting algorithm that assumes it's possible to create an array of buckets and distribute the items to be sorted into those buckets by index.

comparison counting sort and Pigeonhole sort

It is similar to counting sort, but differs in that it "moves items twice: once to the bucket array and again to the final destination [whereas] counting sort builds an auxiliary array then uses the array to compute each item's final destination and move the item there."

Sort Name	Time	Space
Patience Sort	O(n log n)	O(n)
Comb Sort	O(n^2)	O(1)
Counting Sort	O(n+k)	O(MaxValue)
Pigeonhole sort	O(n + Range)	O(n + Range)
Bucket Sort	O(n^2)	O(n+k)

References :

1 - **https://www.geeksforgeeks.org/patience-sorting/**
2 - Patience sorting - Wikipedia
3- Comb Sort - GeeksforGeeks
4- Comb sort - Wikipedia
5- Counting Sort - GeeksforGeeks
6- Counting Sort (With Code in Python/C++/Java/C) (programiz.com)
7- Pigeonhole Sort - GeeksforGeeks
8- Pigeonhole sort - Wikipedia
9- Bucket Sort - GeeksforGeeks
10- Bucket Sort (With Code in Python, C++, Java and C) (programiz.com)