geeksforgeeks.org

Pigeonhole Sort - GeeksforGeeks

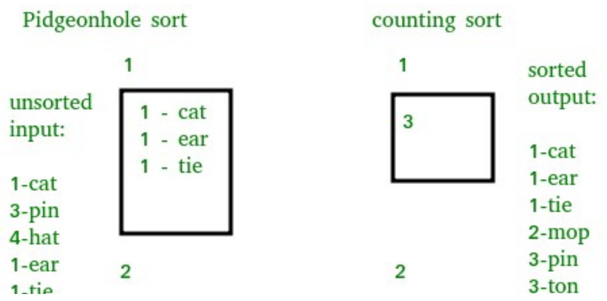
4-5 minutes

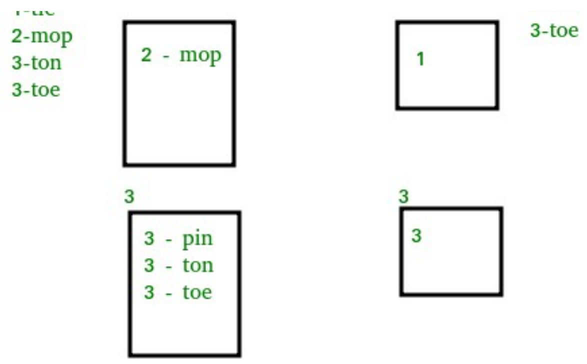
Pigeonhole sorting is a sorting algorithm that is suitable for sorting lists of elements where the number of elements and the number of possible key values are approximately the same.
It requires O(*n* + *Range*) time where n is number of elements in input array and 'Range' is number of possible values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

Working of Algorithm :

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.

Comparison with Counting 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 ".





- C++
- Java
- Python3

C++

```
#include <bits/stdc++.h>
using namespace std;
void pigeonholeSort(int arr[], int n)
{
    int min = arr[0], max = arr[0];
    for (int i = 1; i < n; i++)
    {
        if (arr[i] < min)
            min = arr[i];
        if (arr[i] > max)
            max = arr[i];
    }
    int range = max - min + 1;
    vector<int> holes[range];
    for (int i = 0; i < n; i++)
        holes[arr[i]-min].push_back(arr[i]);
    int index = 0;
    for (int i = 0; i < range; i++)
    {
       vector<int>::iterator it;
       for (it = holes[i].begin(); it !=
holes[i].end(); ++it)
            arr[index++]  = *it;
    }
}
int main()
{
    int arr[] = {8, 3, 2, 7, 4, 6, 8};
    int n = sizeof(arr)/sizeof(arr[0]);
    pigeonholeSort(arr, n);
    printf("Sorted order is : ");
    for (int i = 0; i < n; i++)
        printf("%d ", arr[i]);
    return 0;
}
```

Java

```
import java.lang.*;
import java.util.*;

public class GFG

{

    public static void pigeonhole_sort(int arr[],

                                       int n)

    {

        int min = arr[0];

        int max = arr[0];

        int range, i, j, index;

        for(int a=0; a<n; a++)

        {

            if(arr[a] > max)

                max = arr[a];

            if(arr[a] < min)

                min = arr[a];

        }

        range = max - min + 1;

        int[] phole = new int[range];

        Arrays.fill(phole, 0);

        for(i = 0; i<n; i++)

            phole[arr[i] - min]++;

        index = 0;

        for(j = 0; j<range; j++)

            while(phole[j]-->0)
```

```
                arr[index++]=j+min;
    }
    public static void main(String[] args)
    {
        GFG sort = new GFG();
        int[] arr = {8, 3, 2, 7, 4, 6, 8};
        System.out.print("Sorted order is : ");
        sort.pigeonhole_sort(arr,arr.length);
        for(int i=0 ; i<arr.length ; i++)
            System.out.print(arr[i] + " ");
    }
}
```

Python3

```
def pigeonhole_sort(a):
    my_min = min(a)
    my_max = max(a)
    size = my_max - my_min + 1
    holes = [0] * size
    for x in a:
        assert type(x) is int, "integers only
please"
        holes[x - my_min] += 1
    i = 0
    for count in range(size):
        while holes[count] > 0:
            holes[count] -= 1
            a[i] = count + my_min
            i += 1
a = [8, 3, 2, 7, 4, 6, 8]
print("Sorted order is : ", end = ' ')
pigeonhole_sort(a)
for i in range(0, len(a)):
    print(a[i], end = ' ')
```

Output:

```
Sorted order is : 2 3 4 6 7 8 8
```

Pigeonhole sort has limited use as requirements are rarely met. For arrays where range is much larger than *n*, bucket sort is a generalization that is more efficient in space and time.

References:
https://en.wikipedia.org/wiki/Pigeonhole_sort

This article is contributed **Ayush Govil**. Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above

Other Sorting Algorithms on GeeksforGeeks/GeeksQuiz
Selection Sort, Bubble Sort, Insertion Sort, Merge Sort, Heap Sort, QuickSort, Radix Sort, Counting Sort, Bucket Sort, ShellSort, Comb Sort,

Writing code in comment? Please use ide.geeksforgeeks.org, generate link and share the link here.