

Bucket Sort

```
class TestBucketSort {
    private int size;
    private int[] count;
    TestBucketSort(int size) {
        this.size = size;
        count = new int[size];
    }
    public int getSize() {
        return size;
    }
    public void sort(int[] array) {
        for(int i = 0; i < size; i++)
            count[array[i]]++;
        for(int i = 0, j = 0; i < size; i++)
            while(count[i] > 0) {
                array[j++] = i;
                count[i]--;
            }
    }
    public void show(int[] arr) {
        for(int i = 0; i < arr.length; i++) {
            if(arr[i] > 0) {
                System.out.print("\t" + arr[i]);
                if(i % 9 == 0) System.out.println();
            }
        }
    }
    public static void main(String[] args) {
        TestBucketSort bks = new TestBucketSort(100);
        int list[] = new int[bks.getSize()];
        // for(int i = 0; i < bks.getSize(); i++) {
        //     list[i] = (int)(Math.random() * 100);
        // }
        list[0] = 51;
        list[1] = 46;
        list[2] = 14;
        list[3] = 35;
        list[4] = 22;
        bks.sort(list);
        bks.show(list);
    }
}
```

Code

1. ปัญหาที่อัลกอริทึมเหมาะในการแก้ไขในสถานการณ์ใด

- ในส่วนของ Bucket Sort ที่จำเป็นต่อการใช้ในการแก้ปัญหาคือ ใช้ในการจัดเรียงข้อมูลของ Array ที่มีข้อมูลขนาดเล็กไม่ใหญ่มาก หรือ อาจจะใช้ในตอนที่มีข้อมูลที่เป็น Array กระจัดกระจายไม่เข้าที่

2. Code ที่สำคัญของ Bucket Sort

```
public void sort(int[] array) {
```

for(int i = 0; i < size; i++) ในส่วนนี้จะทำการ วนลูป เพื่อนับข้อมูลจำนวนของ Array

count[array[i]]++; ส่วนนี้คือใช้ Count ในการเป็นตัวแทน Array

```
for(int i = 0, j = 0; i < size; i++)
```

while(count[i] > 0) { ในส่วนนี้จะเป็นการ ถ่ายโอนข้อมูลจากกลุ่มข้างบน มาใช้งาน โดยจะใช้ While /Loop

array[j++] = i; ดึงข้อมูลมา

count[i]--;

```
}
```

```
}
```

อ้างอิง

<http://sci.feu.ac.th/faa/dsa/bookPDFs/chap5-Sorting.pdf>

thaiall.com/class/TestBucketSort.java