

桶排序 (Bucket Sort)

■ 执行流程

- ① 创建一定数量的桶 (比如用数组、链表作为桶)
- ② 按照一定的规则 (不同类型的数据, 规则不同), 将序列中的元素均匀分配到对应的桶
- ③ 分别对每个桶进行单独排序
- ④ 将所有非空桶的元素合并成有序序列

■ 元素在桶中的索引

□ 元素值 * 元素数量

0.34	0.47	0.29	0.84	0.45	0.38	0.35	0.76
------	------	------	------	------	------	------	------

0
1
2
3
4
5
6
7

0.34	0.29	0.35
0.47	0.45	0.38

0.84	0.76
------	------

0
1
2
3
4
5
6
7

0.29	0.34	0.35
0.38	0.45	0.47

0.76	0.84
------	------

0.29	0.34	0.35	0.38	0.45	0.47	0.76	0.84
------	------	------	------	------	------	------	------

桶排序 - 实现

```
double[] array = {0.34, 0.47, 0.29, 0.84, 0.45, 0.38, 0.35, 0.76};
```

```
// 桶数组
List<Double>[] buckets = new List[array.length];
for (int i = 0; i < array.length; i++) {
    int bucketIndex = (int) (array[i] * array.length);
    List<Double> bucket = buckets[bucketIndex];
    if (bucket == null) {
        bucket = new LinkedList<>();
        buckets[bucketIndex] = bucket;
    }
    bucket.add(array[i]);
}
```

```
// 对每个桶进行排序
int index = 0;
for (int i = 0; i < buckets.length; i++) {
    if (buckets[i] == null) continue;
    buckets[i].sort(null);
    for (Double d : buckets[i]) {
        array[index++] = d;
    }
}
```

■ 空间复杂度: $O(n + m)$, m 是桶的数量

■ 时间复杂度: $O(n) + m * O\left(\frac{n}{m} * \log \frac{n}{m}\right) = O\left(n + n * \log \frac{n}{m}\right) = O(n + n * \log n - n * \log m)$

□ 因此为 $O(n + k)$, k 为 $n * \log n - n * \log m$

□ 属于稳定排序