



链滴

日常算法——桶排序

作者: [JackHoo](#)

原文链接: <https://ld246.com/article/1509891009439>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

<h3 id="序言">序言</h3>

<p>时常动动脑，做做算法题还是挺有意思的，温习下桶排序。</p>

<h3 id="桶排序">桶排序</h3>

<p>桶排序是排序算法中最简单最快的排序算法。主要思想就是，以数组为桶，桶有序号，序号即数的索引。遍历待排序的数组，把该数值放入与之相等的序号的桶内，然后从桶中取出非零值得下标即。</p>

<h3 id="代码实现">代码实现</h3>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">public class BucketSort {
</span></span><span class="highlight-line"><span class="highlight-cl">    private int[] buckets;
</span></span><span class="highlight-line"><span class="highlight-cl">    private int[] array;
</span></span><span class="highlight-line"><span class="highlight-cl">    public BucketSort(int range, int[] array) {
</span></span><span class="highlight-line"><span class="highlight-cl">        this.buckets = new int[range];
</span></span><span class="highlight-line"><span class="highlight-cl">        this.array = array;
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">    public void sort() {
</span></span><span class="highlight-line"><span class="highlight-cl">        for (int i = 0; i < array.length; i++) {
</span></span><span class="highlight-line"><span class="highlight-cl">            buckets[array[i]]++;
</span></span><span class="highlight-line"><span class="highlight-cl">        }
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">    public void print() {
</span></span><span class="highlight-line"><span class="highlight-cl">        //desc
</span></span><span class="highlight-line"><span class="highlight-cl">        for (int i = buckets.length - 1; i >= 0; i--) {
</span></span><span class="highlight-line"><span class="highlight-cl">            for (int j = buckets[i]; j >= 0; j--) {
</span></span><span class="highlight-line"><span class="highlight-cl">                System.out.println(i);
</span></span><span class="highlight-line"><span class="highlight-cl">            }
</span></span><span class="highlight-line"><span class="highlight-cl">        }
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span><span class="highlight-line"><span class="highlight-cl">    public static void main(String[] args) {
</span></span><span class="highlight-line"><span class="highlight-cl">        int[] array = {
</span></span><span class="highlight-line"><span class="highlight-cl">            2, 64, 45, 23, 23};
</span></span><span class="highlight-line"><span class="highlight-cl">        BucketSort bucketSort = new BucketSort(65,array);
</span></span><span class="highlight-line"><span class="highlight-cl">        bucketSort.sort();
</span></span><span class="highlight-line"><span class="highlight-cl">        bucketSort.print();
</span></span><span class="highlight-line"><span class="highlight-cl">    }
</span></span></code></pre>
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">}
</span></span></code></pre>
<h3 id="桶排序优点">桶排序优点</h3>
<p>快速简单</p>
<h3 id="桶排序缺点">桶排序缺点</h3>
<p>需要知道待排序最大值，并创建一个这么大的数组，如果最大值为 10w 那就 gg 了，所以桶排
是以空间换时间的方式，适合范围比较小的排序</p>
<h3 id="时间复杂度">时间复杂度</h3>
<p>由于需要遍历所有桶，时间复杂度为  $O(n+m)$ ,  $n$  为待排序元素个数， $m$  为桶的个数。 </p>
```