



黑客派

日常算法——快速排序

作者: [JackHoo](#)

原文链接: <https://hacpai.com/article/1509963564444>

来源网站: [黑客派](#)

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

<h3 id="快速排序">快速排序</h3>

<script async src="https://pagead2.googlesyndication.com/pagead/js/adsbygoogle.js"></script>

<!-- 黑客派PC帖子内嵌-展示 -->

<ins class="adsbygoogle" style="display:block" data-ad-client="ca-pub-5357405790190342" data-ad-slot="8316640078" data-ad-format="auto" data-full-width-responsive="true"></ins>

<script>

(adsbygoogle = window.adsbygoogle || []).push({};

</script>

<p>快速排序采用的思想是分治思想。
快速排序是找出一个元素（理论上可以随便找一个）作基准(pivot)，然后对数组进行分区操作，使基准左边元素的值都不大于基准值，基准右边的元素值不小于基准值，如此作为基准的元素调整到排序后的正确位置。递归快速排序，将其他 n-1 个元素也整到排序后的正确位置。最后每个元素都是在排序后的正确位置，排序完成。所以快速排序算法的核算法是分区操作，即如何调整基准的位置以及调整返回基准的最终位置以便分治递归。
如图：<p>

<p></p>

<h3 id="代码实现">代码实现</h3>

<pre><code class="language-java highlight-chroma">package /span> Suanfa ;

/**

* Created by hushangjie on 2017/11/6.

*/

public class QuickSort {

public static void sort (int arr left int right) {

int left = _left = _left =

int right = _right =

int temp = 0 =

if (left not found render function for node [type=NodeHTMLEntity, Tokens=<]not found render function for node [type=NodeHTMLEntity, Token=<]= right) {


```

highlight-o">=</span> <span class="highlight-n">temp</span><span class="highlight-o">
/span>
<span class="highlight-n">sort</span><span class="highlight-o">      (</span><span c
ass="highlight-n">arr</span><span class="highlight-o">      ,</span> <span class="hi
hlight-n">0</span><span class="highlight-o">      ,</span> <span class="highlight-n">
eft</span> <span class="highlight-o">-</span> <span class="highlight-n">1</span><span
class="highlight-o">);</span>
<span class="highlight-n">sort</span><span class="highlight-o">      (</span><span c
ass="highlight-n">arr</span><span class="highlight-o">      ,</span> <span class="hi
hlight-n">left</span> <span class="highlight-o">+</span> <span class="highlight-n">1</
pan><span class="highlight-o">      ,</span> <span class="highlight-n">_right</span><span
an class="highlight-o">);</span>
<span class="highlight-o">}</span>
<span class="highlight-o">}</span>

<span class="highlight-o">}</span>
</code></pre>

```

特点

- 当分区选取的基准元素为待排序元素中的最大或最小值时，为最坏的情况，时间复杂度和接插入排序的一样，移动次数达到最大值 $C_{max} = 1+2+\dots+(n-1) = n*(n-1)/2 = O(n^2)$ 此时最好间复杂为 $O(n^2)$
- 当分区选取的基准元素为待排序元素中的"中值"，为最好的情况，时间复杂度为 $O(n\log_2n)$ 。
- 快速排序的空间复杂度为 $O(\log_2n)$ 。
- 当待排序元素类似[6,1,3,7,3]且基准元素为 6 时，经过分区，形成[1,3,3,6,7]，两个 3 的位置发生了改变，所以快速排序是一种不稳定排序

```

<script async src="https://pagead2.googlesyndication.com/pagead/js/adsbygoogle.js"></scr
pt>
<!-- 黑客派PC帖子内嵌-展示 -->
<ins class="adsbygoogle" style="display:block" data-ad-client="ca-pub-5357405790190342"
data-ad-slot="8316640078" data-ad-format="auto" data-full-width-responsive="true"></in
>
<script>
  (adsbygoogle = window.adsbygoogle || []).push({});
</script>

```