首先来回顾一下快速排序：

def quicksort(arr, lo, hi)

{

if(lo>=hi)

Return;

int temp = partition(arr, lo, hi)

quicksort(arr, lo, temp-1)

quicksort(arr, temp+1, hi)

}

关于快排，我们已经知道了一般的公式为：

(其中n代表数列的长度，j代表分割点的位置)

若你不太清楚上面公式的由来，请移步：<https://www.cnblogs.com/HDK2016/p/6876313.html>

接下来我们进入正式的分析。通过上面我们知道了一般情况下的公式，但是j的位置是不确定的，所以只能遍历从0到n的所有的位置，假定每个位置的出现的概率是相等的。

……

将全部等式的左边和等式右边分别相加。

接下来利用T(n)减T(n-1)来化简上面这个和式

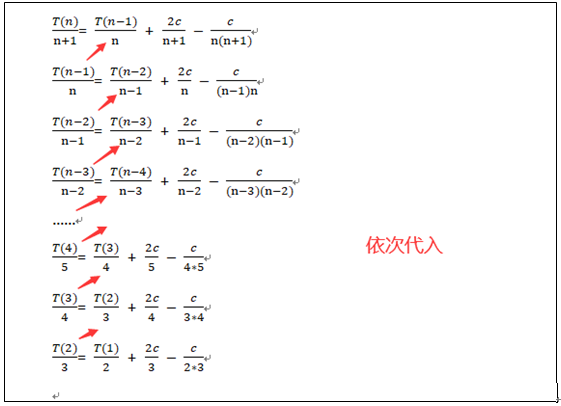
①

②

用①减②，得到下面的表达式：

这个表达式的化简同样是解递归法：

……



由于T(1) = 0,所以：

解到这里，基本上就可以套公式了：

1. 这个是[调和级数.](https://baike.baidu.com/item/%E8%B0%83%E5%92%8C%E7%BA%A7%E6%95%B0/8019971?fr=aladdin) 前n+1部分的和约等于
2. 这个表达式可化简为 ： 最终可得到：。当n趋近于无穷大时，这个表达式的值约等于1。

替换后得：

最高阶为

因此，快排的平均复杂度是O(**)**