https://www.jianshu.com/p/8d0706e7a187

排序算法不仅仅只能用于排序，其稍加改变便可以产生一些意想不到的效果。例如，利用堆排序算法的变体可以快速在无序队列中查找到前某几位或者后某几位元素，利用快速排序算法的变体可以在无序数组中求中位数。

不进行排序，找出序列的中位数。

利用快速排序求无序数组的中位数的步骤如下：

* 步骤一、选出一个元素作为无序数组的分割元素，并将这个元素与首元素交换位置。
* 步骤二、向后遍历这个数组，比较元素与分割元素的大小，如果元素比分割元素小，则先将元素与分割元素的前一位调换位置，再将其与分割元素调换位置。
* 步骤三、完成遍历后，数组以分割元素的界限，分成两个无序数组，前面的数组比后面的数组都要小。
* 步骤四、如果分割元素的位置比（n+1）/2要大，则以前面的无序数组重新按照步骤一执行，如果小于，则以后面的元素按照步骤一执行。一直到所选择的分割元素的位置为（n+1）/2为止，此分割元素即为中位数。
* 整个算法时间复杂度为O(n)。通过上述的搜索代码，去查找有一百万元素的无序数组，发现其平均遍历次数为三百五十万次，并且比较不稳定。经过研究发现不稳定的原因为：分割元素的选择。我们每次选择分割元素是随机选择的，分割元素与中位数总是有一定的距离误差，如果误差比较大时，下一次需要遍历的数组就大。因此，需要对分割点的选择做优化。

优化果无序数组个数大于100，先随机选择30个元素。

* 求出这三十个元素的平均值。
* 求出最接近平均值的元素。
* 将这个元素作为分割点。