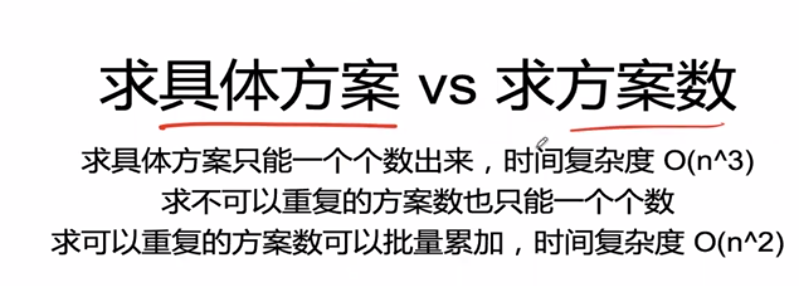
相向双指针：时间复杂度为。

哈希表中的元素是无序的。

做算法题时，首先将算法的运行图画出来。

三数之和必须要降维，因为两数之和容易理解。

lintcode 57中三数之和等于指定的target，如果求具体的所有方案，时间复杂度为。但lintcode 482中的3数中任意两数之和大于第3个数，如果求具体方案，最坏的时间复杂度为，例如如果数组全是1,[1, 1, 1, 1, 1, 1]，任意3个数都满足要求，所有方案数为。



在算法分析中，要有降维的思想，如果要求，就应该想到for循环一个n，然后用二分法，堆等时间复杂度为的算法；如果要求，for循环一个n，然后用双指针算法等。

k数之和有两种考法：



由于k的不确定性，无法对应到两数之和或者3数之和的情况，碰到求“方案总数”情况，要立刻反应过来用DP，动态规划，碰见求“具体方案”，要反应过来使用DFS，深度优先搜索。

相向双指针：分割数组，in-place(原地排序，不使用额外空间)



partition相比于快速排序，要求选中的枢纽元，必须在右边，而快速排序并不要求，它希望尽可能的均匀划分。时间复杂度为。

时间复杂度与最内层的循环主体的执行次数有关，与多少重循环无关。

