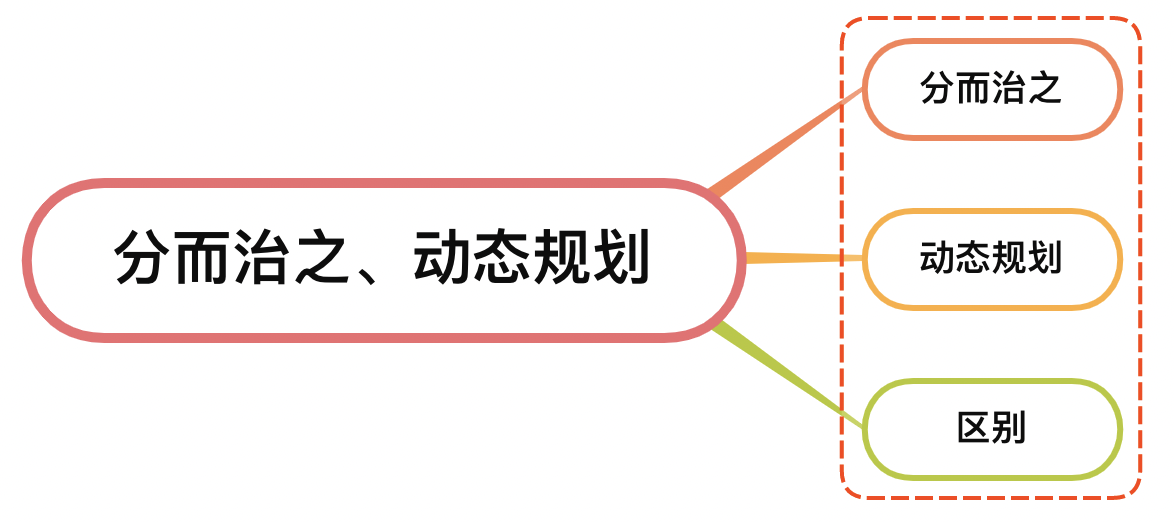
# 面试官：说说你对分而治之、动态规划的理解？区别？



## 一、分而治之

分而治之是算法设计中的一种方法，就是把一个复杂的问题分成两个或更多的相同或相似的子问题，直到最后子问题可以简单的直接求解，原问题的解即子问题的解的合并

关于分而治之的实现，都会经历三个步骤：

* 分解：将原问题分解为若干个规模较小，相对独立，与原问题形式相同的子问题
* 解决：若子问题规模较小且易于解决时，则直接解。否则，递归地解决各子问题
* 合并：将各子问题的解合并为原问题的解

实际上，关于分而治之的思想，我们在前面已经使用，例如归并排序的实现，同样经历了实现分而治之的三个步骤：

* 分解：把数组从中间一分为二
* 解决：递归地对两个子数组进行归并排序
* 合并：将两个字数组合并称有序数组

同样关于快速排序的实现，亦如此：

* 分：选基准，按基准把数组分成两个字数组
* 解：递归地对两个字数组进行快速排序
* 合：对两个字数组进行合并

同样二分搜索也能使用分而治之的思想去实现，代码如下：

function binarySearch(arr,l,r,target){  
 if(l> r){  
 return -1;  
 }  
 let mid = l + Math.floor((r-l)/2)  
 if(arr[mid] === target){  
 return mid;  
 }else if(arr[mid] < target ){  
 return binarySearch(arr,mid + 1,r,target)  
 }else{  
 return binarySearch(arr,l,mid - 1,target)  
 }  
}

## 二、动态规划

动态规划，同样是算法设计中的一种方法，是一种在数学、管理科学、计算机科学、经济学和生物信息学中使用的，通过把原问题分解为相对简单的子问题的方式求解复杂问题的方法

常常适用于有重叠子问题和最优子结构性质的问题

简单来说，动态规划其实就是，给定一个问题，我们把它拆成一个个子问题，直到子问题可以直接解决

然后呢，把子问题答案保存起来，以减少重复计算。再根据子问题答案反推，得出原问题解的一种方法。

一般这些子问题很相似，可以通过函数关系式递推出来，例如斐波那契数列，我们可以得到公式：当 n 大于 2的时候，F(n) = F(n-1) + F(n-2) ，

f(10)= f(9)+f(8),f(9) = f(8) + f(7)...是重叠子问题，当n = 1、2的时候，对应的值为2，这时候就通过可以使用一个数组记录每一步计算的结果，以此类推，减少不必要的重复计算

### 适用场景

如果一个问题，可以把所有可能的答案穷举出来，并且穷举出来后，发现存在重叠子问题，就可以考虑使用动态规划

比如一些求最值的场景，如最长递增子序列、最小编辑距离、背包问题、凑零钱问题等等，都是动态规划的经典应用场景

关于动态规划题目解决的步骤，一般如下：

* 描述最优解的结构
* 递归定义最优解的值
* 按自底向上的方式计算最优解的值
* 由计算出的结果构造一个最优解

## 三、区别

动态规划算法与分治法类似，其基本思想也是将待求解问题分解成若干个子问题，先求解子问题，然后从这些子问题的解得到原问题的解

与分治法不同的是，适合于用动态规划求解的问题，经分解得到子问题往往**不是互相独立**的，而分而治之的子问题是相互独立的

若用分治法来解这类问题，则分解得到的子问题数目太多，有些子问题被重复计算了很多次

如果我们能够保存已解决的子问题的答案，而在需要时再找出已求得的答案，这样就可以避免大量的重复计算，节省时间

综上，可得：

* 动态规划：有最优子结构和重叠子问题
* 分而治之：各子问题独立

## 参考文献

* https://baike.baidu.com/item/%E5%8A%A8%E6%80%81%E8%A7%84%E5%88%92/529408
* https://juejin.cn/post/6951922898638471181