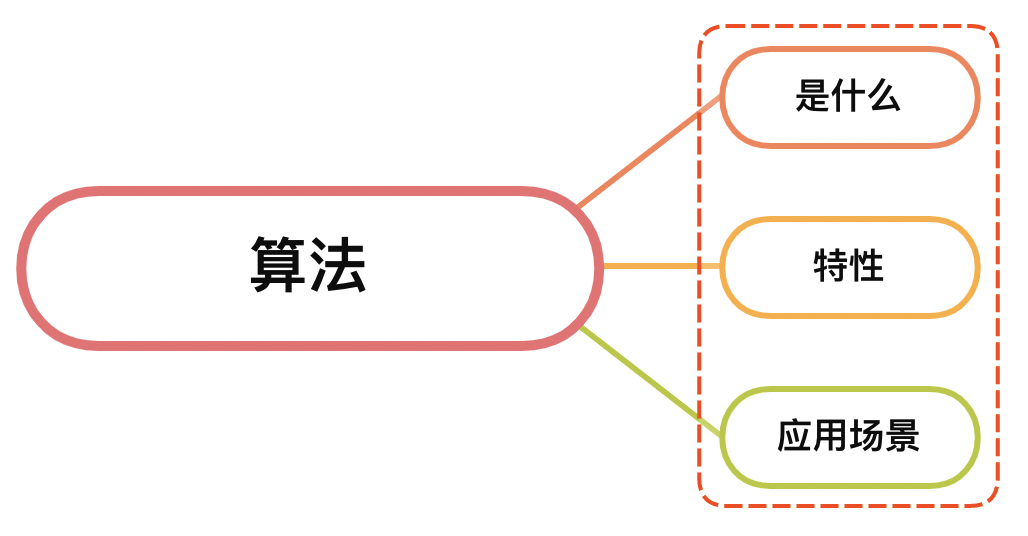
# 面试官：说说你对算法的理解？应用场景？



## 一、是什么

算法（Algorithm）是指解题方案的准确而完整的描述，是一系列解决问题的清晰指令，算法代表着用系统的方法描述解决问题的策略机制

也就是说，能够对一定规范的输入，在有限时间内获得所要求的输出

如果一个算法有缺陷，或不适合于某个问题，执行这个算法将不会解决这个问题

一个程序=算法+数据结构，数据结构是算法实现的基础，算法总是要依赖于某种数据结构来实现的，两者不可分割

因此，算法的设计和选择要同时结合数据结构，简单地说数据结构的设计就是选择存储方式，如确定问题中的信息是用数组存储还是用普通的变量存储或其他更加复杂的数据结构

针对上述，可以得出一个总结：不同的算法可能用不同的时间、空间或效率来完成同样的任务

## 二、特性

关于算法的五大特性，有如下：

* 有限性（Finiteness）：一个算法必须保证执行有限步之后结束
* 确切性（Definiteness）： 一个算法的每一步骤必须有确切的定义
* 输入（Input）：一个算法有零个或多个输入，以刻画运算对象的初始情况，所谓零个输入是指算法本身给定了初始条件
* 输出（Output）：一个算法有一个或多个输出。没有输出的算法毫无意义
* 可行性（Effectiveness）：算法中执行的任何计算步骤都是可以被分解为基本的可执行的操作步骤，即每个计算步骤都可以在有限时间内完成（也称之为有效性）

## 三、应用场景

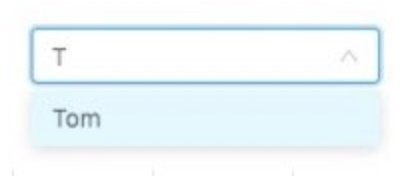
在前端领域中，数据结构与算法无法不在，例如现在的vue或者react项目，实现虚拟DOM或者Fiber结构，本质就是一种数据结构，如下一个简单的虚拟DOM：

{  
 type: 'div',  
 props: {  
 name: 'lucifer'  
 },  
 children: [{  
 type: 'span',  
 props: {},  
 children: []  
 }]  
}

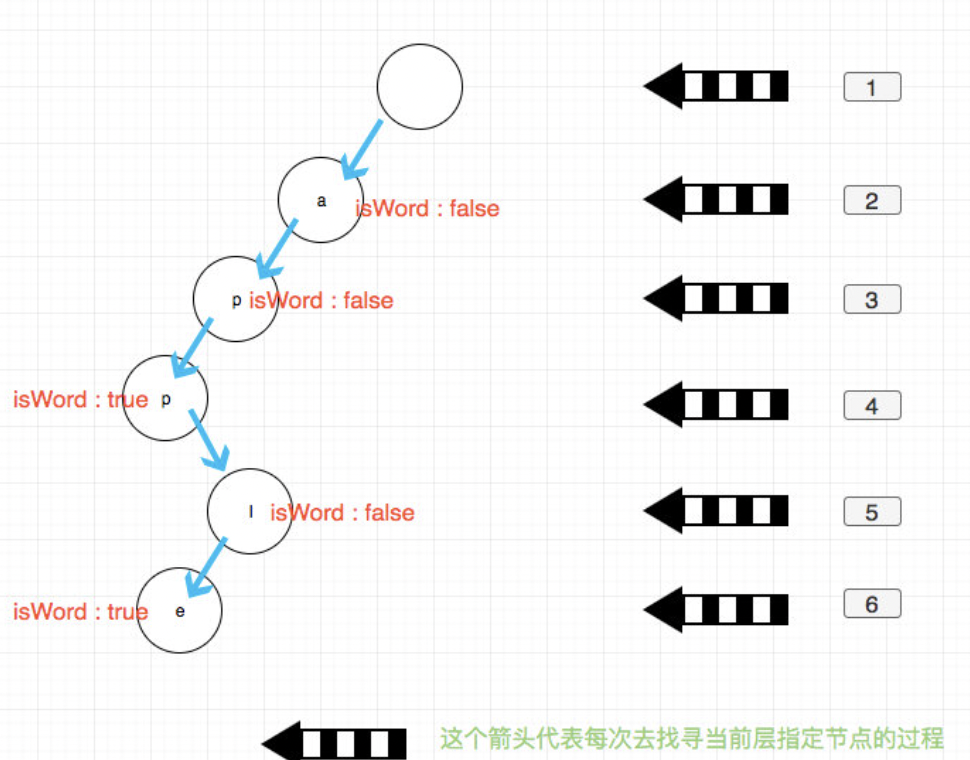
vue与react都能基于基于对应的数据结构实现diff算法，提高了整个框架的性能以及拓展性

包括在前端javascript编译的时候，都会生成对应的抽象语法树AST，其本身不涉及到任何语法，因此你只要编写相应的转义规则，就可以将任何语法转义到任何语法，也是babel， PostCSS, prettier， typescript

除了这些框架或者工具底层用到算法与数据结构之外，日常业务也无处不在，例如实现一个输入框携带联想功能，如下：



如果我们要实现这个功能， 则可以使用前缀树，如下：



包括前端可能会做一些对字符串进行相似度检测，例如"每日一题"和"js每日一题"两个字符串进行相似度对比，这种情况可以通过“最小编辑距离”算法，如果a和b的编辑距离越小，我们认为越相似

日常在编写任何代码的都需要一个良好的算法思维，选择好的算法或者数据结构，能让整个程序效率更高

## 参考文献

* https://baike.baidu.com/item/%E7%AE%97%E6%B3%95/209025
* https://lucifer.ren/blog/2019/09/18/algorthimn-fe-1/