

## 如何高效实现英文单词前缀匹配

- 有一个英文单词库，数组，有几十万个英文单词。
- 当我输入一些字母时，判断已经输入的字母是不是被包含在单词库中，或单词库中某一单词的前缀。

方法一：

将单词库暴力的遍历一次，使用 `String.prototype.includes()` 来判断单词库中某一个单词是否包含输入的字符串，或单词库中某一个单词的前缀是不是该字符串。

时间复杂度是  $O(n)$ ，而且 `includes()` 还有自己的时间复杂度，总时间复杂度是大于  $O(n)$ 。

方法二：

- 将单词库先进行拆分，以 a 字母开头的放一起，以 b 字母开头的放一起，依此方式，拆分成 26 个不同的单词库了。
- 再循环上面操作，将 a 开头的单词库再进行拆分，又拆分成以 a 开头，以 b 开头等，再拆分成 26 份。

比如一个数组：`['abc', 'abs', 'boss', 'base']`。

拆分成：

```
const obj = {
  a: {
    b: {
      s: {},
      c: {}
    }
  },
  b: {
    o: {
      s: {
        s: {}
      }
    },
    a: {
      s: {
        e: {}
      }
    }
  },
}
```

获取单词的时候通过对象的 key 来获取。比如：`obj.a.b.s`。

拆分的时候有点复杂，但是单词库在一段时间内是固定的，不需要频繁拆分，拆分一次可以用很久。拆分之后时间复杂度就很小了。 $O(1)$ ， $O(1)$  乘以单词字符长度（单词最长10多个字符），时间复杂度不会很大。