1. 先看例子：

S=aaabbbbbbaabaaba

1. 先把单个字符的分配代码，a的代码是0，b的代码是1。（实际编码的时候用整个ASCII码就行，0-255都用上，举例：’A’的代码是65）
2. 字符串和编码的映射放入字典中（key，value）。Key是字符串，value是对应的代码。现在有（a，0）（b，1）
3. 用c代表当前读到的字符，用p代表c之前存在于字典中的字符串。读第一个字符串的时候p是空的。
4. 开始读字符串，读入a，c=a，p为空，p和c组成的字符串（现在是a）在字典中有吗?有，p=（p和c）。下一步再读一个字符，c=a，p和c组成的字符串（现在是aa）在字典中有吗？没有，输出p（其实是对应的代码），并将（2，aa）插入到字典中，且p=c。如此下去。如果读到字尾，输出p（的代码）就可以了。（遇到字典中没有的数对，我才把当前c前面的p输出，遇到的都是字典中有的数对，我就一直让p做最大匹配）

下表是读取S=aaabbbbbbaabaaba的过程，每一次读取一个字符，直到读到空字符

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 读到的第i个字符 | p | c | 输出（实际中输出的是相应的代码，我放在括号里） | 新加入字典中的数对 |
| 1 | 空 | a |  |  |
| 2 | a | a | a(0) | (2,aa) |
| 3 | a | A |  |  |
| 4 | aa | b | aa(2) | (3,aab) |
| 5 | b | b | b(1) | (4,bb) |
| 6 | b | b |  |  |
| 7 | bb | b | bb(4) | (5,bbb) |
| 8 | b | b |  |  |
| 9 | bb | b |  |  |
| 10 | bbb | a | bbb(5) | (6,bbba) |
| 11 | a | a |  |  |
| 12 | aa | b |  |  |
| 13 | aab | a | aab(3) | (7,aaba) |
| 14 | a | a |  |  |
| 15 | aa | b |  |  |
| 16 | aab | a |  |  |
| 17 | aaba | 空 | aaba(7) |  |

所以S编码是0214537

再来看字典：

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 关键字（key） | a | b | aa | aab | bb | bbb | bbba | aaba |
| 代码（value） | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

代码越大，关键字越长，越往后不是越多…

解决方法是一部分关键字用代码代替，aa用0a代替，aab用2b代替，用代码+一个字节的形式，这样可以如2b我可以追溯到0ab，再往上追随到aab。我们程序中每个代码用12位表示，最多分配4096个代码，那么一个关键字是12+8=20位。S的编码0214537需要12\*7=84位，压缩了吧！

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 关键字（key） | a | b | 0a | 2b | 1b | 4b | 5a | 3a |
| 代码（value） | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

奇怪，关键字a和0a是怎么区分的呢，我认为是区分不了的，编解码的时候当然可以根据0-255是一个字符，超过255就是2个字符来区分，我知道。我的意思是map中关键字a和0a是一样的关键字，同一个map中没有2个一样的key。不过因为0是null，实际并不会有问题，null和一下个字符还不是只有一个字符，还是解释为代码0。当然这个0是null只是特例，书中这个例子我觉得还稍稍有点瑕疵。