

1、JDK 1.7及之前，为什么采用头插法

呃... 这个可能需要问头插法的实现者了；

但有种说法，我觉得挺有道理：缓存的时间局部性原则，最近访问过的数据下次大概率会再次访问，把刚访问过的元素放在链表最前面可以直接被查询到，减少查找次数

2、既然头插法有链表成环的问题，为什么直到 1.8 才采用尾插法来替代头插法

只有在并发情况下，头插法才会出现链表成环的问题，多线程情况下，HashMap 本就非线程安全，这就相当于你在它的规则之外出了问题，那能怪谁？

1.8 采用尾插，是对 1.7 的优化

、

3、既然 1.8 没有链表成环的问题，那是不是说明可以把 1.8 中的 HashMap 用在多线程中

链表成环只是并发问题中的一种，1.8 虽然解决了此问题，但是还是会有很多其他的并发问题，比如：上秒 put 的值，下秒 get 的时候却不是刚 put 的值；因为操作都没有加锁，不是线程安全的

怎么扩容的？扩容rehash的流程（jdk 1.7）

-单线程 -多线程

<https://www.jianshu.com/p/13c650a25ed3>

<https://www.cnblogs.com/williamjie/p/9360370.html>

头插-尾插

<https://zhuanlan.zhihu.com/p/342061904>

<https://www.cnblogs.com/youzhibing/p/13915116.html>