

3.1.6 用输入中的单词总数 W 和不同单词总数 D 的函数给出FrequencyCounter调用的put()和get()方法的次数

- 对于put

1. 若key不存在, 则put进
所有不同单词总数第一次put时, 属于这种情况, 一共 D 次
2. 若key已存在, put更新val
这些属于重复输入, 一共 $W - D$ 次
3. 为了寻找最大值, put进” ” 一共1次

三种情况共计

$$D + W - D + 1 = W$$

- 对于get()

1. 对于已经存在的key, 会get其val并更新
这种会产生 $W - D$ 次
2. 在for(String word: st.key())循环中产生的get
循环一共持续size(keys)次, 即 $W - D$ 次, 而其中判断语句if(st.get(word) < st.get(max))一共两次get, 所以一共 $2(W - D)$ 次
3. 输出时get(max)
1次

共计

$$3 * (W - D) + 1$$

3.1.7 对于 $N=10, 10^2, 10^3, 10^4, 10^5, 10^6$, 在 N 个小于1000的随机非负整数中FrequencyCounter平均能够找到多少个不同的键。

3.1.8 在《双城记》中, 使用频率最高的长度大于等于10的单词是什么?
monseigneur 101

3.1.9 在FrequencyCounter中添加追踪put()方法的最后一次调用的代码。打印出最后插入的那个单词以及在此之前总共从输入中处理了多少单词。用你的程序处理tale.txt中长度分别大于等于1,8和10的单词。

for循环里面直接加一个计数器i和临时变量Key lastKey;

3.1.10 给出用SequentialSearchST 将键E A S Y Q U E S T I O N 插入一个空符号表的过程的轨迹。一共进行了多少次比较。

3.1.11 给出用BinarySearchST 将键E A S Y Q U E S T I O N 插入一个空符号表的过程的轨迹。一共几星了多少次比较?

3.1.12 修改BinarySearchST, 用一个Item对象的数组而非两个平行数组来保存键和值。添加一个构造函数, 接受一个Item的数组为参数并将其归并排序。

令 $y = \cos(x)$, 则积分变为

$$\sigma_a$$