Word Frequency in a Book

February 1, 2018

1 Question

Design a method to find the frequency of occurrences of any given word in a book.

2 Explanation and Algorithm

The best way to do this is perform preprocessing on the book. We can create a hash table which maps from a word to its frequency. The frequency of any word can be easily looked up in O(1) time. You must run through the entire book to run through every word, each time updating the hash table.

3 Hints

1. We need to run through the entire book to keep track of every word and it's frequency. We'll need to do pre-processing.
2. What can we use to store these words and their frequency?
3. We can run through the book in O(n) and word frequency look up in O(1).

4 Code

```
HashMap<String, Integer> preprocessDictionary(String [] book){
    HashMap<String,Integer> wordFreqMap = new HashMap<String, Integer>();
    for(String word: book){
        word = word.toLowerCase();
        if(!wordFreqMap.containsKey(word)){
            wordFreqMap.put(word, 1);
            continue;
        }
        int freq = wordFreqMap.get(word);
        freq++;
        table.put(word, freq);
    }
return wordFreqMap;
}
int getWordFreq(Hashtable<String, Integer> bookWordFreqMap, String word){
    if(bookWordFreqMap == null || word ==null)
        return 0;
    word = word.toLowerCase();
    if(bookWordFreqMap.containsKey(word)){
        return bookWordFreqMap.get(word);
    }
    return 0;
}
```

5 Big-O Analysis

You running through preprocessing in O(N) times. And lookup is O(1).

6 Source

Taken from Cracking the Coding Interview by Gayle Laakmann McDowell.