

Homework 10: Hashtables

cs230
Spring 2002

Allen B. Downey
Computer Science Department

Due: Monday 22 April

The purpose of this assignment is to implement a `Hashtable` and some of the algorithms that use one. Please start by reading Chapter 19.

Check those lists

Write a method for the `LinkedList` class that checks whether the list contains a loop. Your method should be linear in the number of nodes.

Random words

Write a method named `randomWord` in the `WordCount` class. It should choose and return a random word from the hashtable, such that the probability that each word appears is in proportion to its frequency in the original document. Hints:

1. It might help to keep track of n , the total number of words in the document. Then you can choose a random integer in the range $(0, n)$.

```
// randInt: returns a random number between low and high,  
// including both endpoints  
public static int randInt (int low, int high) {  
    // the while loop is necessary to cover the  
    // possibility that random returns exactly 1.0  
    while (true) {  
        int x = (int)(Math.random() * (high-low+1) + low);  
        if (x >= low && x <= high) return x;  
    }  
}
```

2. As you traverse the keys in the hashtable, keep a running sum of the frequencies you have seen. When this total exceeds your random number, you've found your word.

Implement that Table

Write a hashtable implementation of the Table ADT (as defined in Section 19.2). Test it by using it with any of the programs you have written so far. Hints:

1. Start with the `Vector` implementation in the book and make sure it works with existing programs.
Modify the implementation to use either an array of `LinkedLists` or a `Vector` of `LinkedLists`, whichever you prefer.
2. To implement `keys`, put all the keys in a vector and then invoke `elements` on the vector.