MATH 182: Homework #6

Professor Wotao Yin

Eric Chuu

Assignment: 3.2.3, 3.2.6, 3.3.2, 3.3.4, 3.3.13

UID: 604406828

March 3, 2017

Exercise 3.4.4

Write a program to find values a, m with m as small as possible, such that the hash function (a * k) mod m for transforming the kth letter of the alphabet into table index produces distinct values for keys S E A R C H X M P L.

Solution. We first convert the keys to their corresponding numeric values as the kth letter of the alphabet. The keys then become

$$[18, 4, 0, 17, 2, 7, 10, 12, 15, 11]$$

The first step is to find values of a, m so that we can obtain a unique encoding (no collisions). This is accomplished in the first loop below. After finding values a, m that give us this property, we then minimize m. This is done accomplished in the second loop. The results were: m = 19, a = 23. □

```java
import java.util.*;

public class HashFunction {
  public static void findPerfectHash(int[] k) {
    ArrayList<Integer> hCodes = new ArrayList<Integer>();
    int a = 1, m = k.length, iter = 0, result;
    while(hCodes.size() != k.length && iter < 100) {
      hCodes.clear();
      for (int i = 0; i < k.length; i++) {
        result = (k[i] * a) % m;
        if (hCodes.contains(result))
          break;
        hCodes.add(result);
      }
      iter++; a += 2; m++;
    }
    iter = 0;
    int last_m = 0;
    ArrayList<Integer> goodHash = new ArrayList<Integer>();
    // minimize m, while fixing k
    while(hCodes.size() == k.length && iter < 100) {
      last_m = m;
      goodHash.clear(); goodHash.addAll(hCodes);
      hCodes.clear();
      m--;
      for (int i = 0; i < k.length; i++) {
        result = (k[i] * a) % m;
        if (!hCodes.contains(result))
          hCodes.add(result);
      }
      iter++;
    }
    System.out.println("m = " + last_m + ", a = " + a);
  }

  public static void main(String[] args) {
    int[] input = {18, 4, 0, 17 ,2, 7, 10, 12, 15, 11};
    findPerfectHash(input);
  }
}
```