Atakan SEVİNÇLİ

TEDU CMPE 343 HW-I PART-II

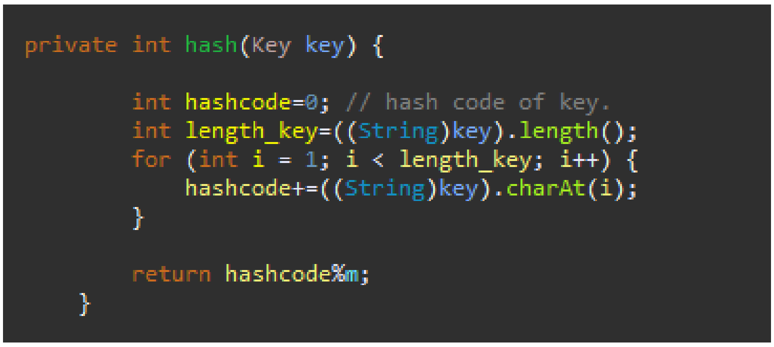
Introduction

In this part of the hw1 we try different hash functions and calculate the statistics for our hash functions.

Average Number of Probes

First of all, we calculate the efficency of given hash function in part 1 and we find average number of probes are 92,556.

The Given Hash Function
Average = 92.556



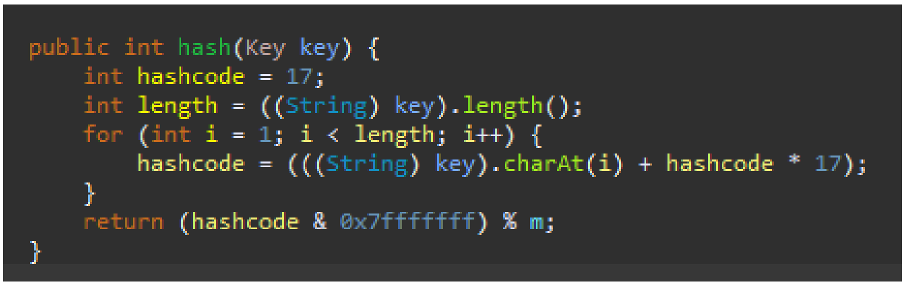
```
private int hash(Key key) {

        int hashcode=0; // hash code of key.
        int length_key=((String)key).length();
        for (int i = 1; i < length_key; i++) {
            hashcode+=((String)key).charAt(i);
        }

        return hashcode%m;
    }
```

Secondly, we calculate the efficency of second hash function that given in the our text book which is Algorithms, 2th Edition by Robert Sedgewick and Kevin Wayne

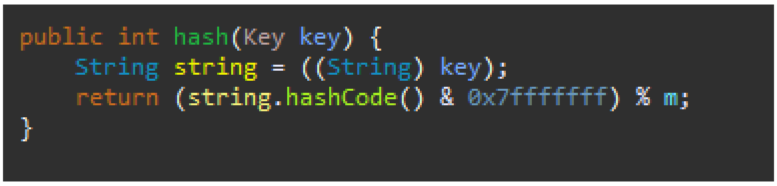
Text Book
Average = 55.660



```
public int hash(Key key) {
    int hashcode = 17;
    int length = ((String) key).length();
    for (int i = 1; i < length; i++) {
        hashcode = (((String) key).charAt(i) + hashcode * 17);
    }
    return (hashcode & 0x7fffffff) % m;
}
```

Finally, we use hash function in java libraries and we find average number of probes as 55,676

Java String hashCode
Average = 55.676



```
public int hash(Key key) {
    String string = ((String) key);
    return (string.hashCode() & 0x7fffffff) % m;
}
```

- ## Conclusion

To sum up, the given hash function in the part 1 is less efficient than other two hash functions and average number probes are highly different than other two function. Other two function is more efficient and more successful alternatives.Basically hash function allow us to make a logical distribution and prevents unlogical collision. Distinctive features of objects are important to us and a good function makes a good distinction.