Department of Computer Engineering

University of Peradeniya

CO 322 Data Structures and Algorithms

Lab 03 - Hash Tables

Name : A.L.V.H.Dharmathilaka

Registration Number : E/16/086

In this lab a hash table is implemented using the method called open or external hashing. It takes Strings (words in the text) as keys. Each bucket would have a link to a linked list which contains the actual key/value stored. When storing a key/value pair, the hash function would select a bucket and add a new node to the list with the new key/value pair. Difference in Hash table is that not like arrays, the index of the bucket is not similar to the hash code(bucket Number). Hash function is used to calculate the hash code. So the hash function should minimize the collisions . Below different hash functions are tried with different text files and different table sizes.

Mainly three different functions are used. They are,

1. Modified method hash = (hash + s.charAt(i)) % M) ------->additional function
2. Division method h(k)=k mod n
3. Multiplication method h(k)=n(K.A mod 1)

Hash functions assume that the keys are natural numbers. So Strings are converted to a natural number using some radix notation in methods.

For the compare purpose two different text files are used .

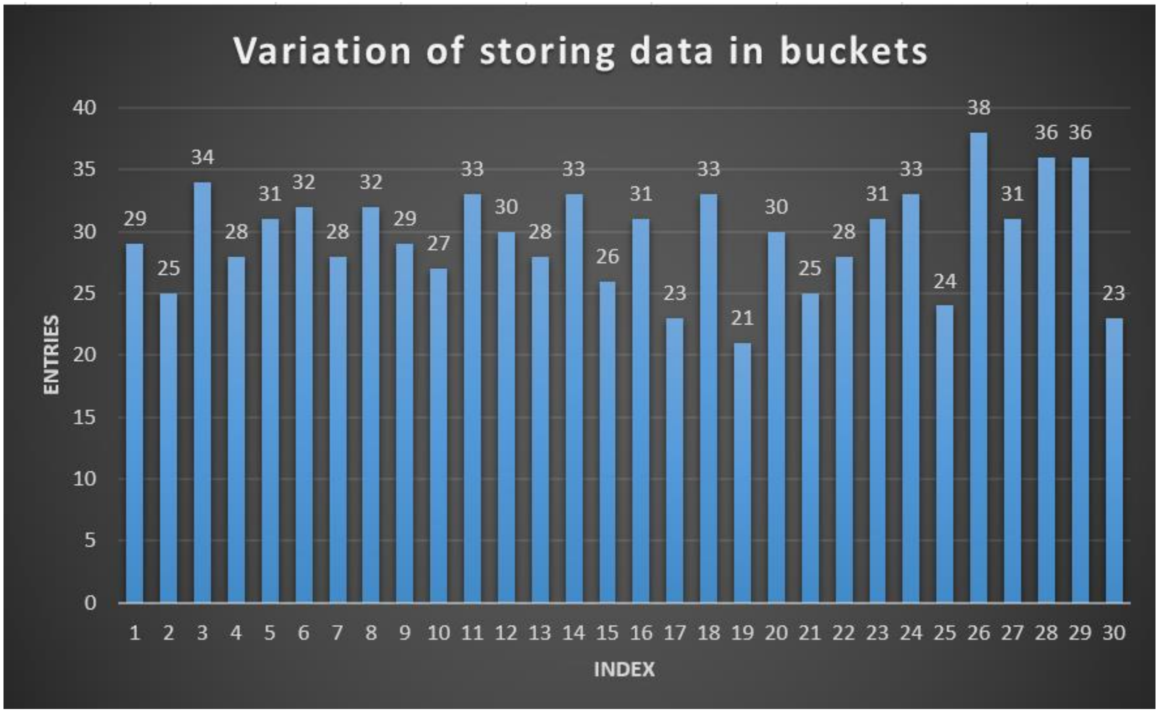
01. modified function

```
private int hashfun(String key) {
        int hash = 0;
        for (int i = 0; i < key.length(); i++){
              hash = ( hash + ((key.charAt(i)))*(int)Math.pow(128,(key.length()-1-i)))%(table.length);
        }
        return (hash%(table.length));
  }
```

In this method though text file, table size changed all buckets are filled in uniform way. That uniformity is clearly shown when the table size increasing. As a example if table size is 50, average number of entries in a bucket is 21.Minimum number of entries in a bucket is14 and Maximum number of entries in a bucket 28.So the standard deviation also very low. When the bucket size is small (example : table size 16) the standard deviation is comparatively large.

Read the file sample-text1.txt

Table size 30



Total entries :888

Average number of entries in a bucket : 29.6

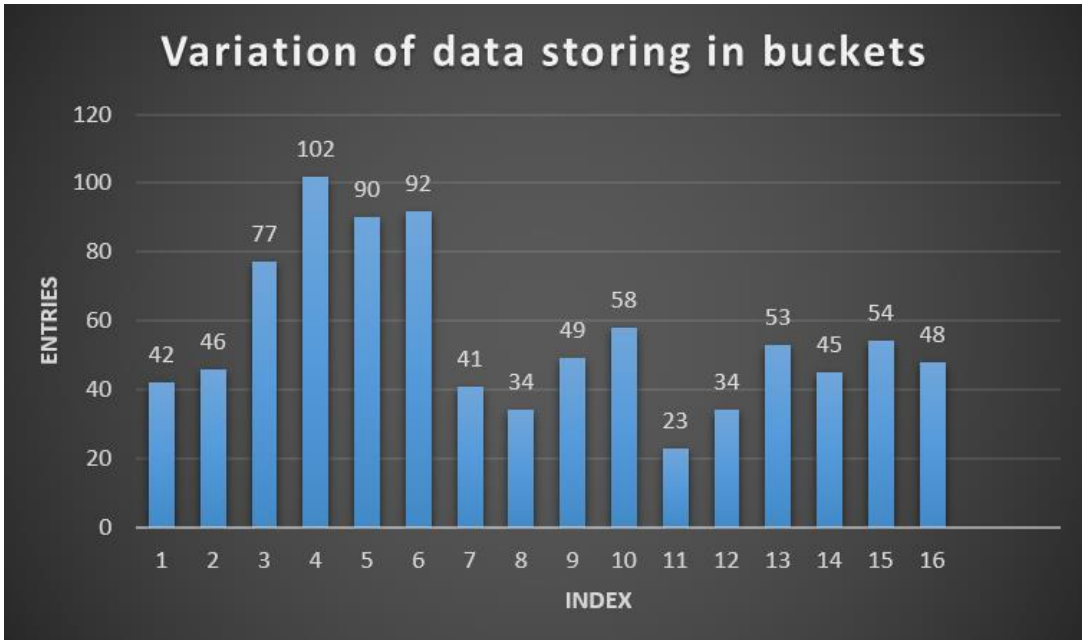
Standard Deviation : 3.9799497484264696

Minimum number of entries in a bucket : 21

Maximum number of entries in a bucket : 38

Read the file sample-text1.txt

Table size 16



Total entries :888

Average number of entries in a bucket : 55.5

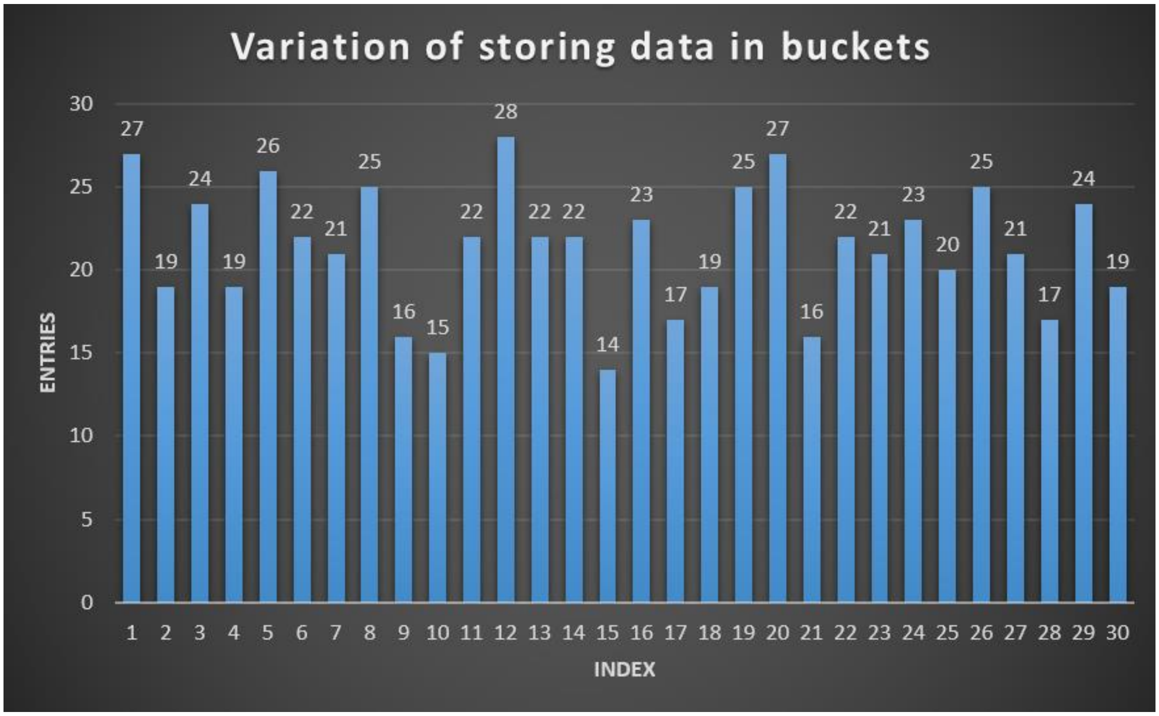
Standard Deviation : 22.107690969434145

Minimum number of entries in a bucket : 23

Maximum number of entries in a bucket : 102

Read the file sample-text2.txt

Table size 30



Total entries :641

Average number of entries in a bucket : 21.366666666666667

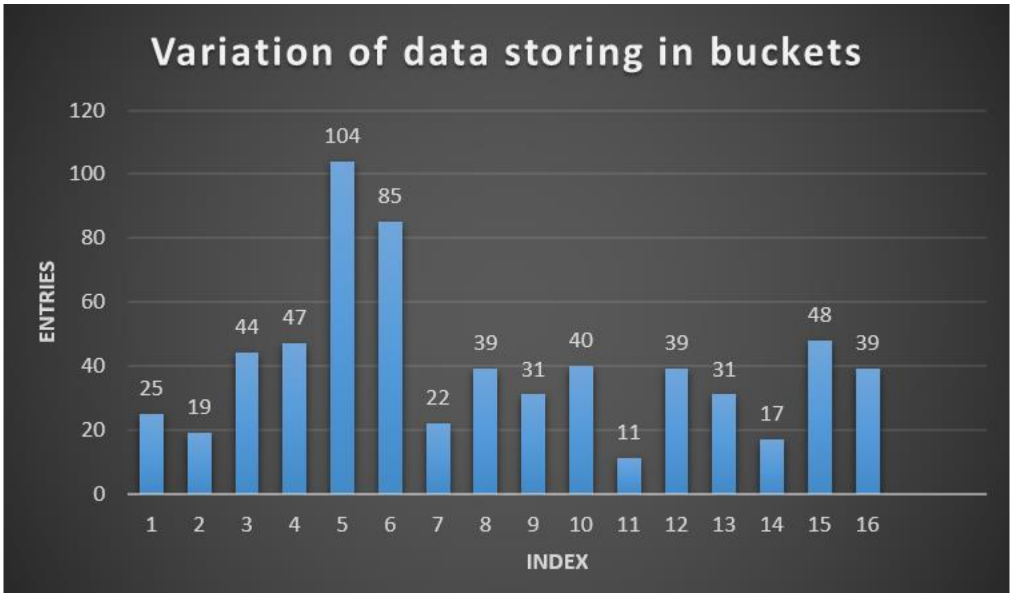
Standard Deviation : 3.6695443253291695

Minimum number of entries in a bucket : 14

Maximum number of entries in a bucket : 28

Read the file sample-text2.txt

Table size 16



Total entries :641

Average number of entries in a bucket : 40.0625

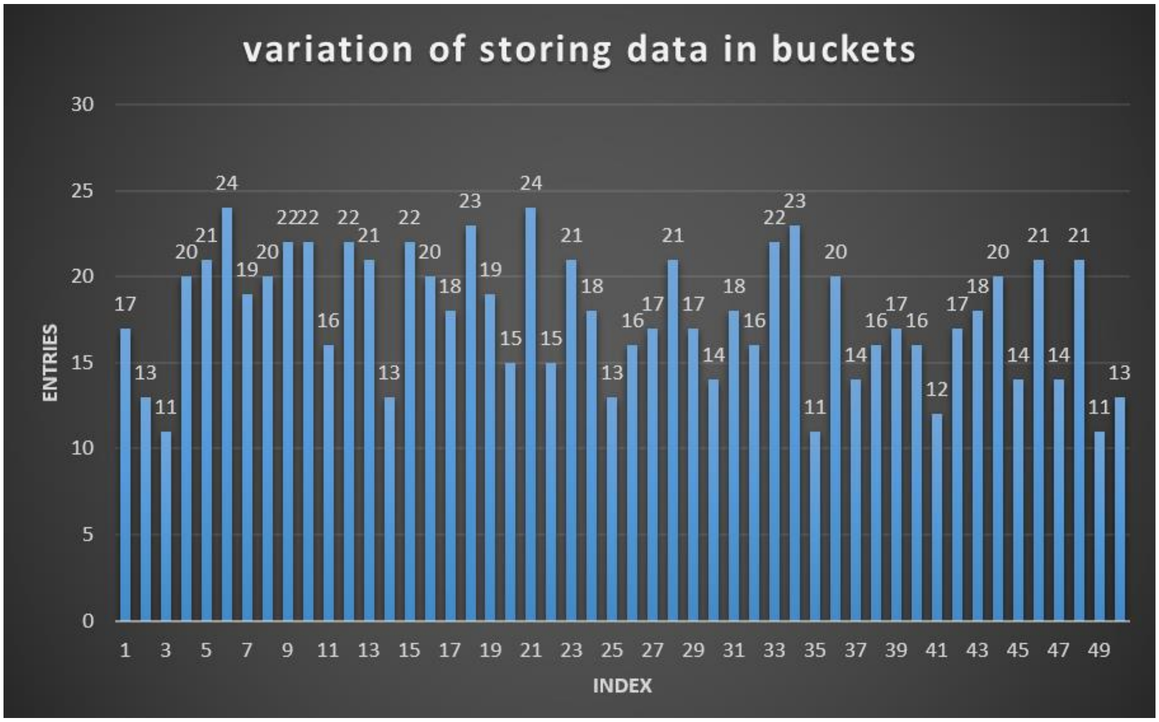
Standard Deviation : 23.409316387925557

Minimum number of entries in a bucket : 11

Maximum number of entries in a bucket : 104

Read the file sample-text1.txt

Table size 50



Total entries :888

Average number of entries in a bucket : 17.76

Standard Deviation : 3.547167884383251

Minimum number of entries in a bucket : 11

Maximum number of entries in a bucket : 24

02. **division method**

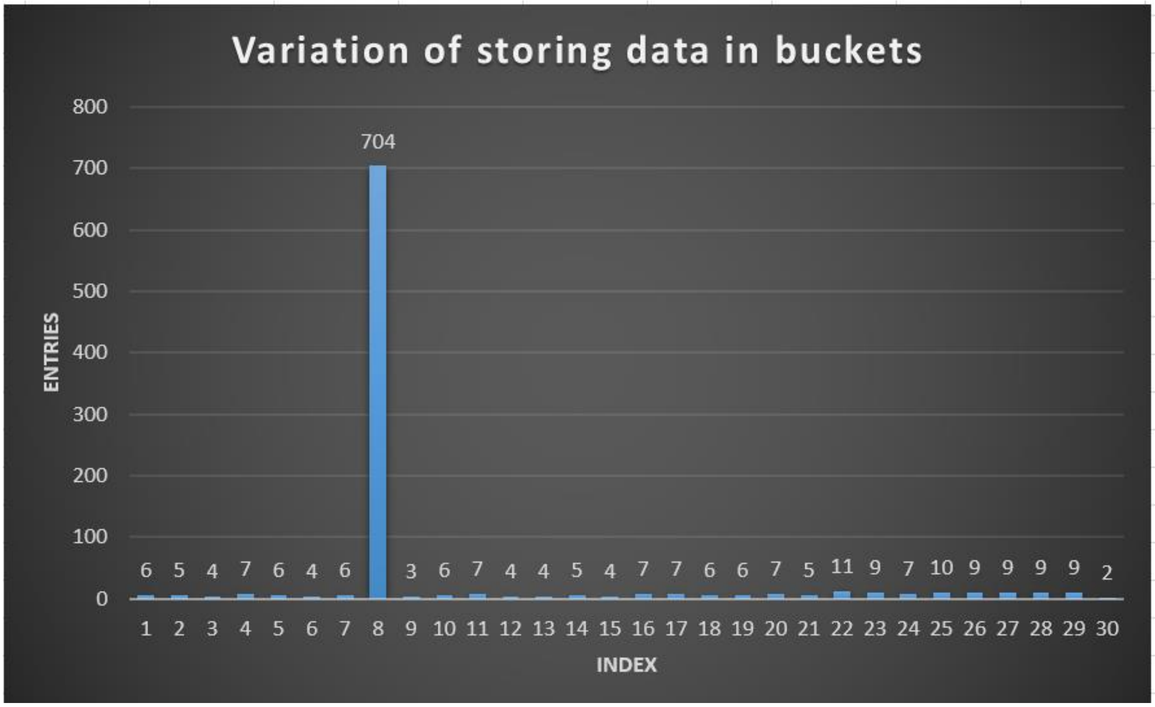
A popular hash function for quick hashing is division method

```
//get ascii value of a String
// h(k)=k mod n where n is the table size
public int getASCII(String word){
        int  sum =0;
        for(int j=0;j< word.length();j++){
                sum+= ((int)word.charAt(j))*(Math.pow(128,(word.length()-1-j)));
        }
        return sum;
}
//Division method
public int divFun(String s){
        int key = getASCII(s);
        return(key%table.length);


}
```

But this method is not much suitable when table size is close to a power of two., because regularity that one see in data is that all the lower order bits are same and higher bits differ or vice versa. Here when table size in 16 =2^4 this method gives a large deviation. For the sample-text1.txt gives 167.04116259173963 standard deviation when table size is 16.

Read the file sample-text1.txt

Table size 30



Total entries :888

Average number of entries in a bucket : 29.6

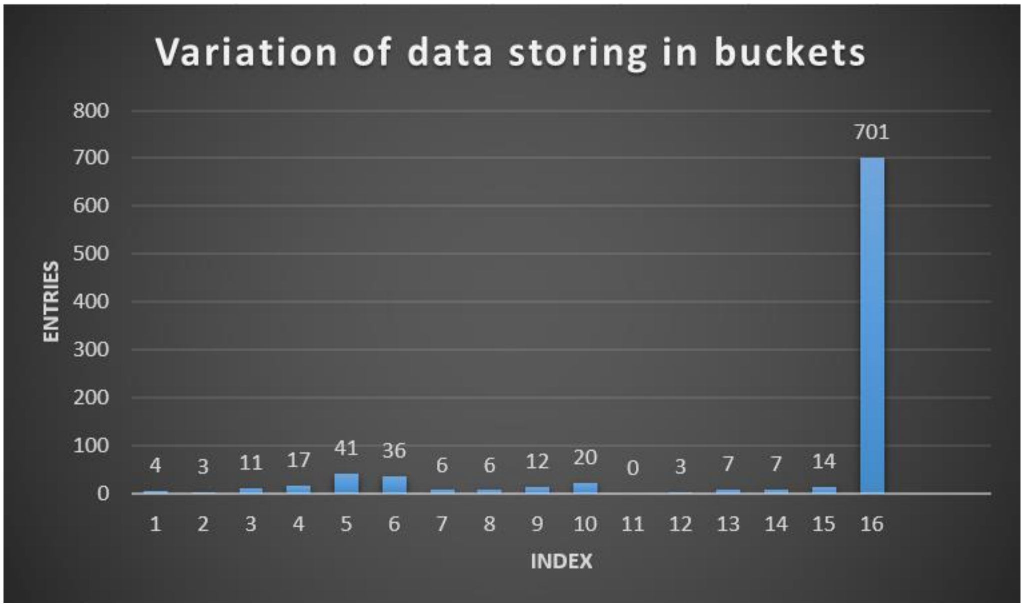
Standard Deviation : 125.25110777953223

Minimum number of entries in a bucket : 2

Maximum number of entries in a bucket : 704

Read the file sample-text1.txt

Table size 16



Total entries :888

Average number of entries in a bucket : 55.5

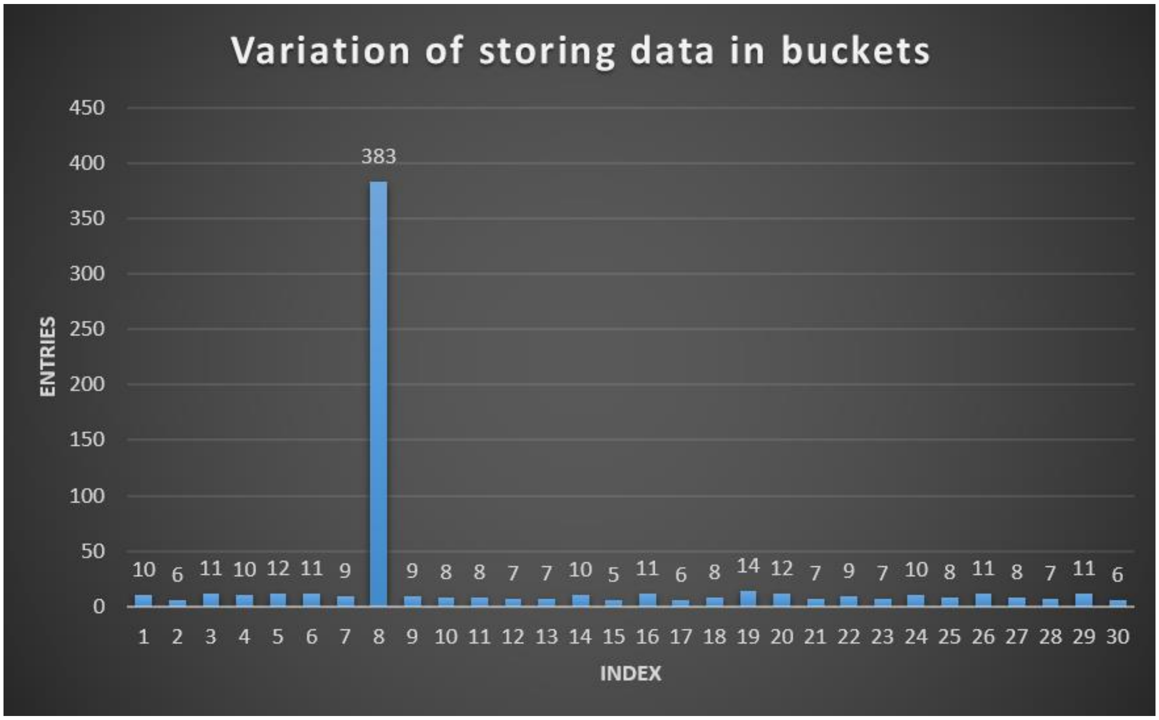
Standard Deviation : 167.04116259173963

Minimum number of entries in a bucket : 0

Maximum number of entries in a bucket : 701

Read the file sample-text2.txt

Table size 30



Total entries :641

Average number of entries in a bucket : 21.366666666666667

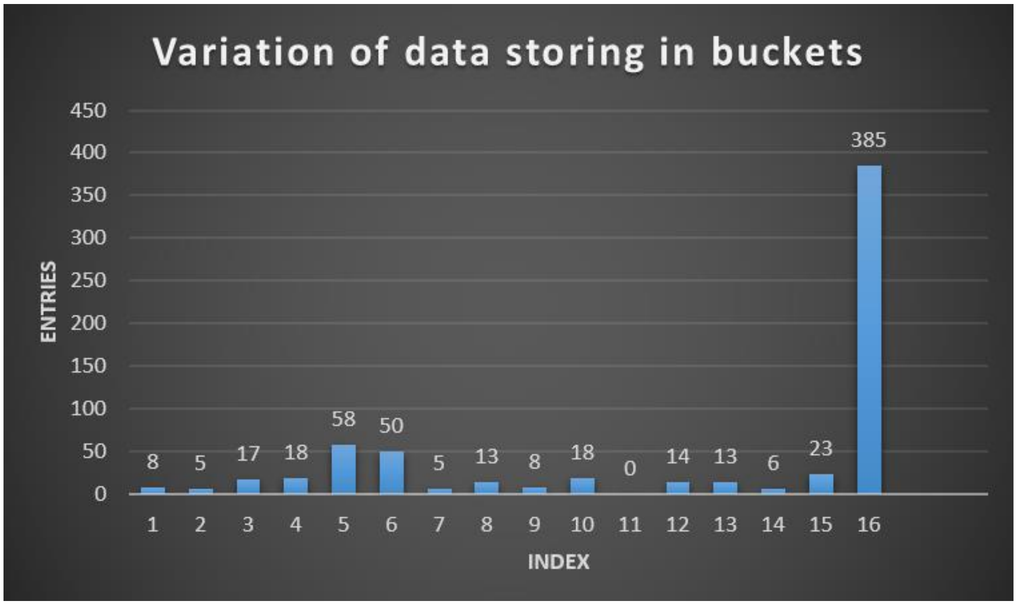
Standard Deviation : 67.18233067969253

Minimum number of entries in a bucket : 5

Maximum number of entries in a bucket : 383

Read the file sample-text2.txt

Table size 16



Total entries :641

Average number of entries in a bucket : 40.0625

Standard Deviation : 90.34930046076727

Minimum number of entries in a bucket : 0

Maximum number of entries in a bucket : 385

02. Multiplication Method

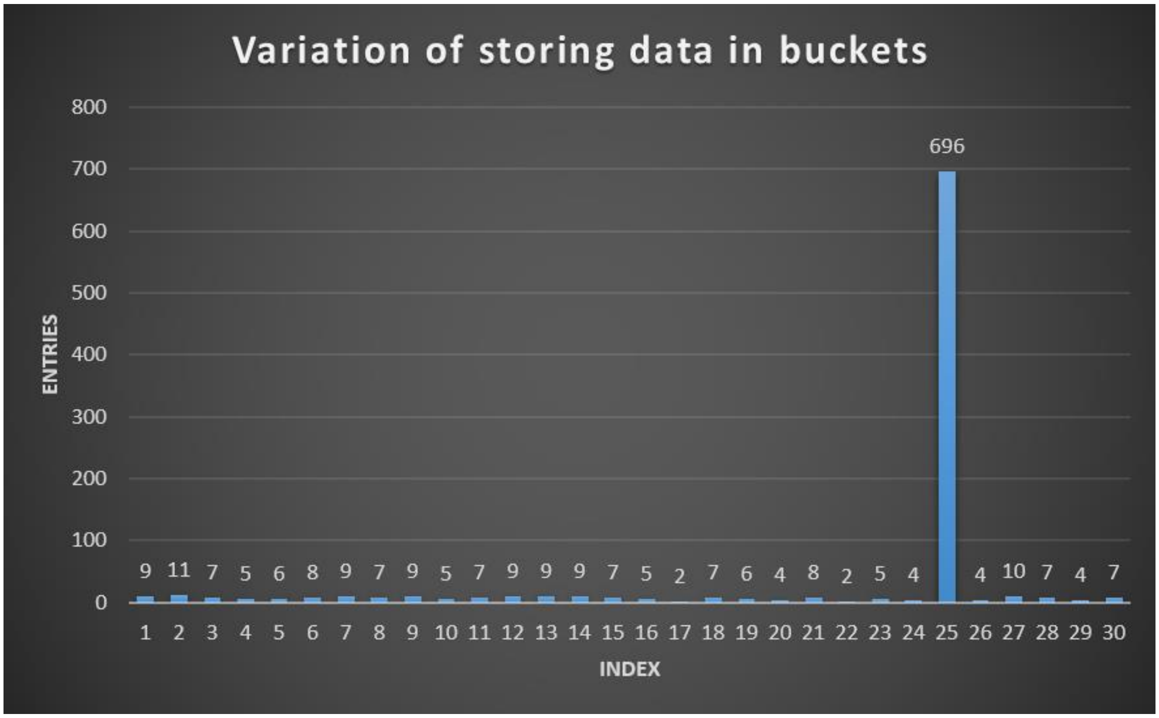
```
//get ascii value of a String
        public int getASCII(String word){
                int  sum =0;
                for(int j=0;j< word.length();j++){
                        sum+= ((int)word.charAt(j))*(Math.pow(128,(word.length()-1-j)));
                }
                return sum;
        }

        //multiplication method
        //h(k)=n( K.A mod 1)
        public int multiFun(String s, double valueA){
                double k = valueA;
                int key = getASCII(s);
                return (int) (table.length * (k * key % 1));
        }
```

Read the file sample-text1.txt

Table size 30

A= 0.618033

h(k)=n(K.A mod 1)



Total entries :888

Average number of entries in a bucket : 29.6

Standard Deviation : 123.76526168517562

Minimum number of entries in a bucket : 2

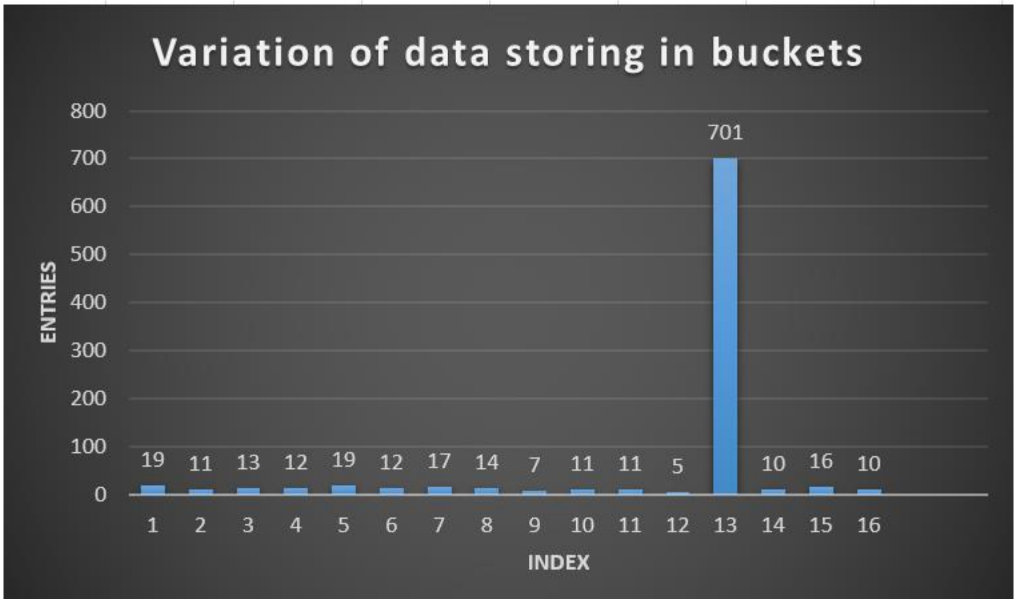
Maximum number of entries in a bucket : 696

Read the file sample-text1.txt

Table size 16

A= 0.618033

h(k)=n(K.A mod 1)



Total entries :888

Average number of entries in a bucket : 55.5

Standard Deviation : 166.70857806363776

Minimum number of entries in a bucket : 5

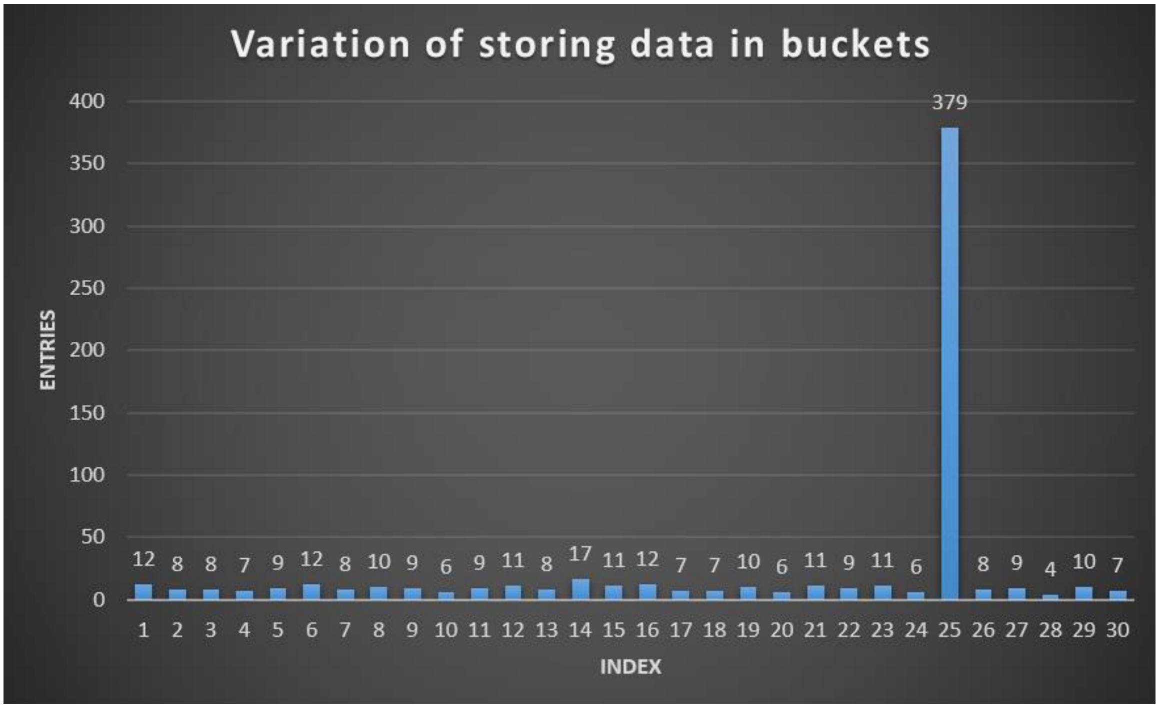
Maximum number of entries in a bucket : 701

Read the file sample-text2.txt

Table size 30

A= 0.618033

h(k)=n(K.A mod 1)



Total entries :641

Average number of entries in a bucket : 21.366666666666667

Standard Deviation : 66.45649370494621

Minimum number of entries in a bucket : 4

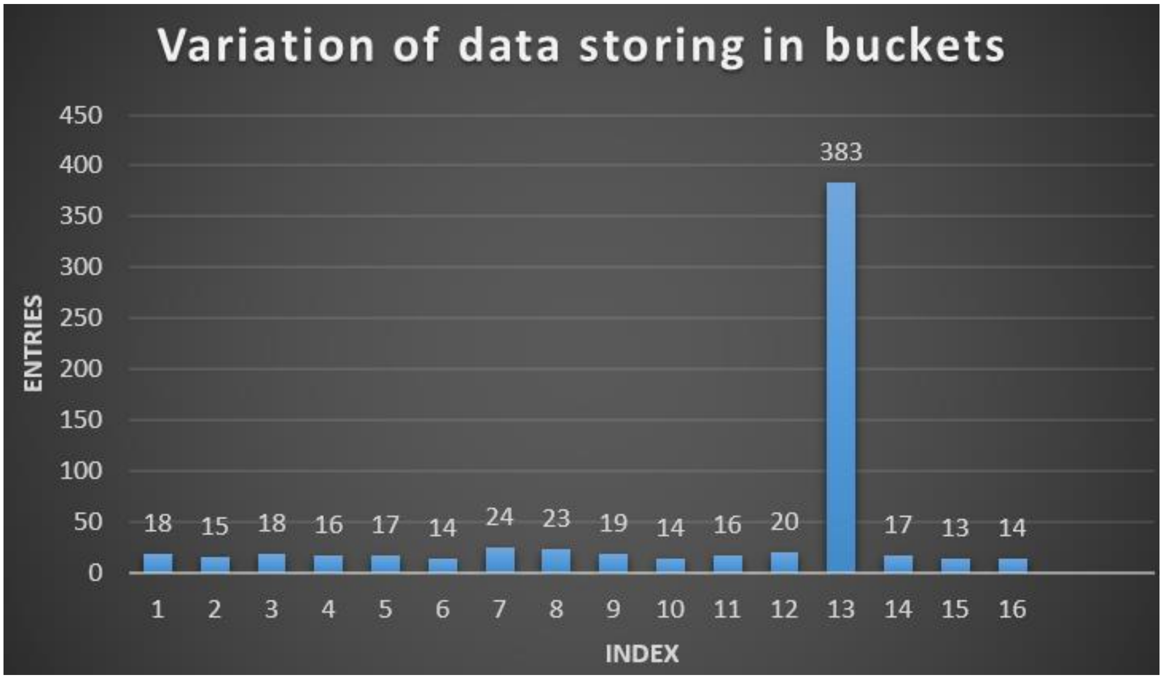
Maximum number of entries in a bucket : 379

Read the file sample-text2.txt

Table size 16

A= 0.618033

h(k)=n(K.A mod 1)



Total entries :641

Average number of entries in a bucket : 40.0625

Standard Deviation : 88.59456018148067

Minimum number of entries in a bucket : 13

Maximum number of entries in a bucket : 383

Comparison table

Method	Hash Table size	Minimum entries	Maximum entries	Average entries	Standard deviation	Text file No.	Total entries
1	30	21	38	29.6	3.979949	1	888
1	16	23	102	55.5	22.107690	1	888
1	30	14	28	21.3	3.669544	2	641
1	16	11	104	40.0625	23.40931	2	641
1	50	11	24	17.76	3.547167	1	888
2	30	2	704	29.6	125.251107	1	888
2	16	0	701	55.5	167.041162	1	888
2	30	5	383	21.366	67.182330	2	641
2	16	0	385	40.0625	90.349300	2	641
3	30	2	696	29.6	123.765261	1	888
3	16	5	701	55.5	166.708578	1	888
3	30	4	379	21.366	66.4564	2	641
3	16	13	383	40.0625	88.594560	2	641

Both multiplication and deviation method has lager standard deviation than the first method. Among them largest standard deviation of size 30 table is 167.0411 it is when division method table size has a power of 2 ($16=2^4$). But no such affect in multiplication method. Also in divison method maximum and large number of entries in a bucket has biggest difference compare to other two methods. And also division method has empty buckets so it is a memory wastage. Considering all these aspects Multiplication method is better than division method because it has less collisions than division method.

Considering the first and 3rd methods , 3rd method has low standard deviation .So it has the minimum collisions. Comparatively it has good uniform distribution than other two methods too. So modified method is better than multiplication method.