

CO322 LAB-03

HASHTABLE

KARIKRAN.V
E/16/172

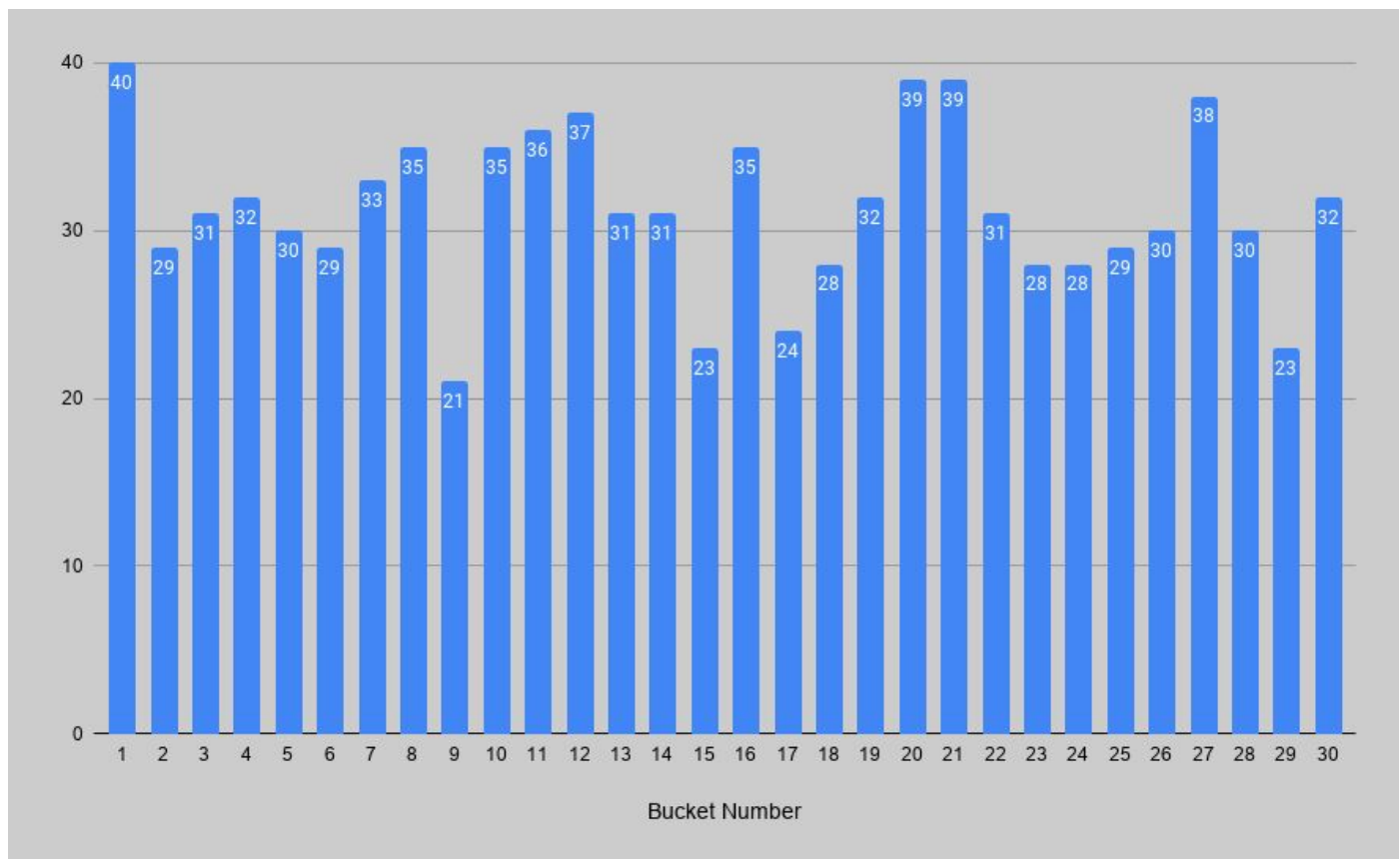
1) For sample-text1.txt file

a) Hash Code 1

private int hash(String key)

```
{  
    int h = 0;  
    for (int i = 0; i < key.length(); i++)  
    {  
        h = (31* h + key.charAt(i))%Table.length;  
    }  
    return (h%Table.length);  
}
```

i) For Hash Code 1 when bucket size 30



Total:939

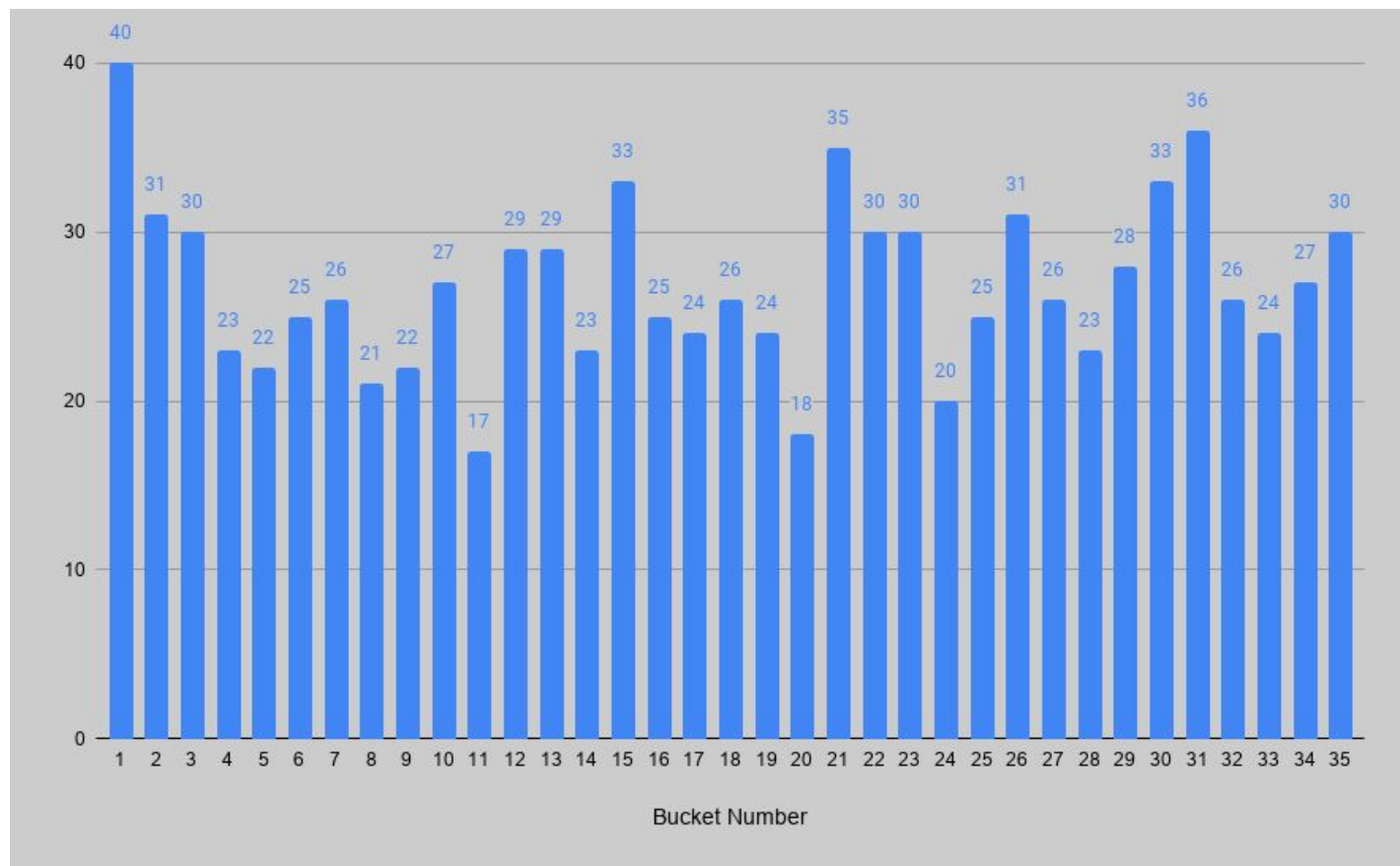
Average:31.3

Deviation:4.723352474011441

Minimum number of entries in buckets:21

Maximum number of entries in buckets:40

i) For Hash Code 1 when bucket size 35



Total:939

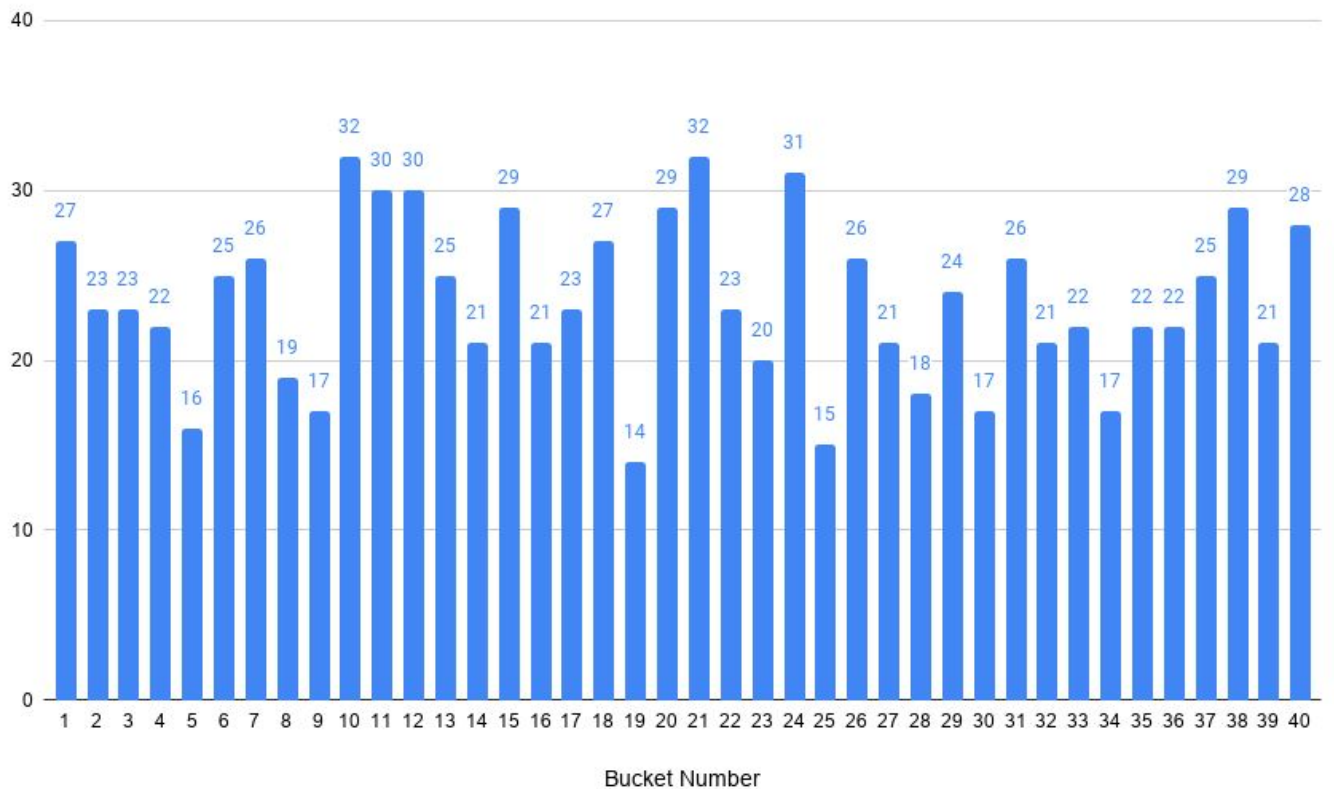
Average:26.828571

Deviation:4.922172610049855

Minimum number of entries in buckets:17

Maximum number of entries in buckets:40

iii) For Hash Code 1 when bucket size 40



Total:939

Average:23.475

Deviation:4.6823474285241

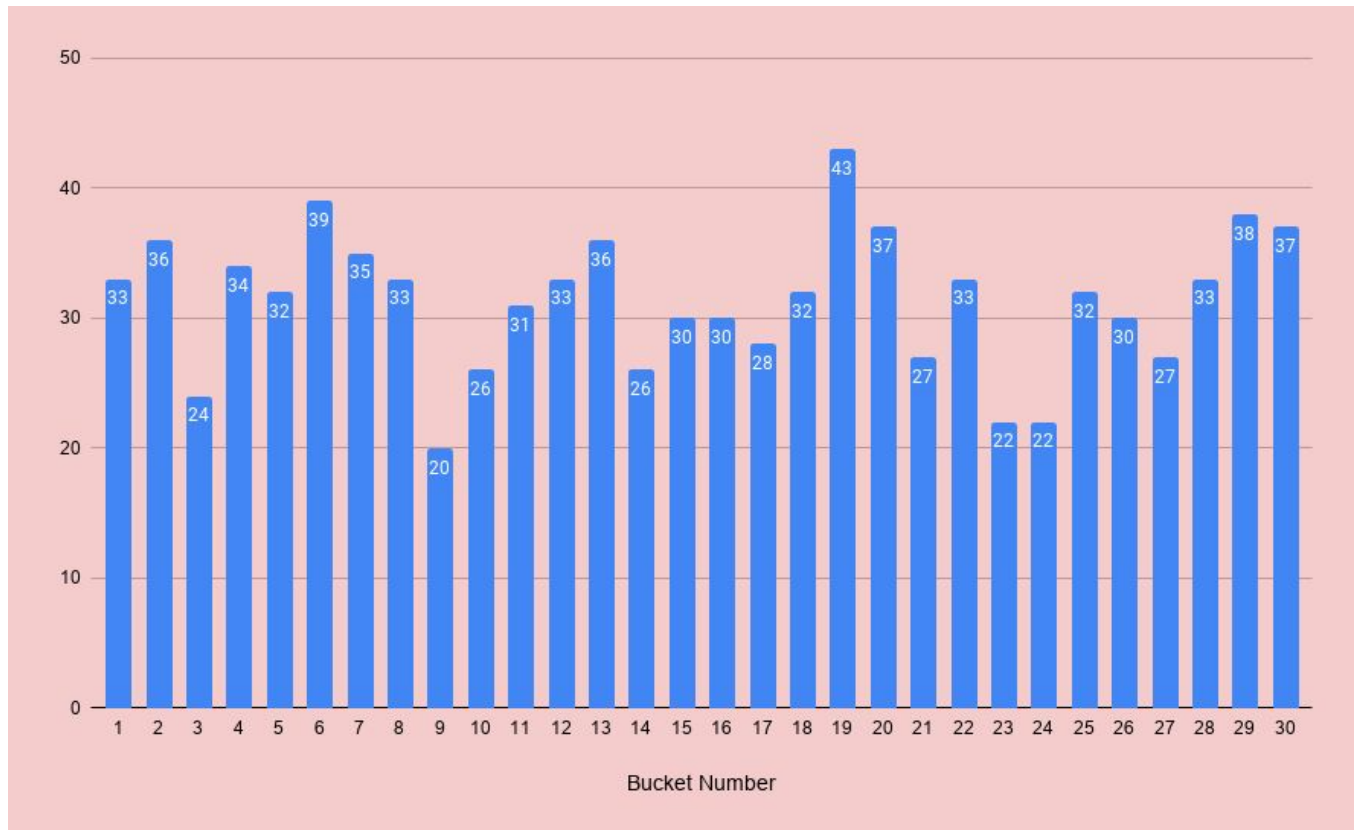
Minimum number of entries in buckets:14

Maximum number of entries in buckets:32

b) Hash Code 2

```
private int hash(String key)
{
    int h = 0;
    for (int i = 0; i < key.length(); i++)
    {
        h = (89* h + key.charAt(i))%Table.length;
    }
    return (h%Table.length);
}
```

i) For Hash Code 2 when bucket size 30



Total:939

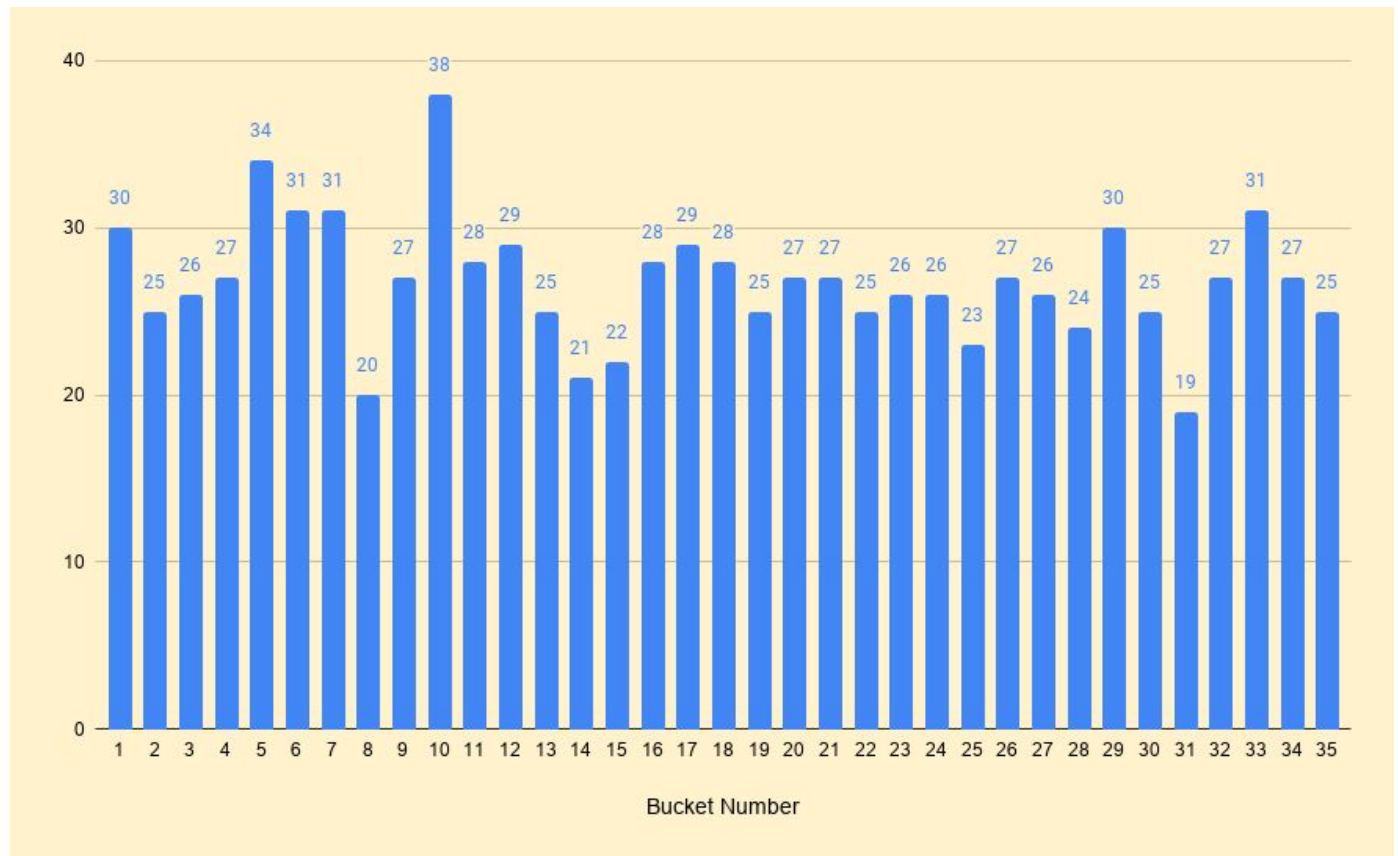
Average:31.3

Deviation:5.320719743958518

Minimum number of entries in buckets:20

Maximum number of entries in buckets:43

ii) For Hash Code 2 when bucket size 35



Total:939

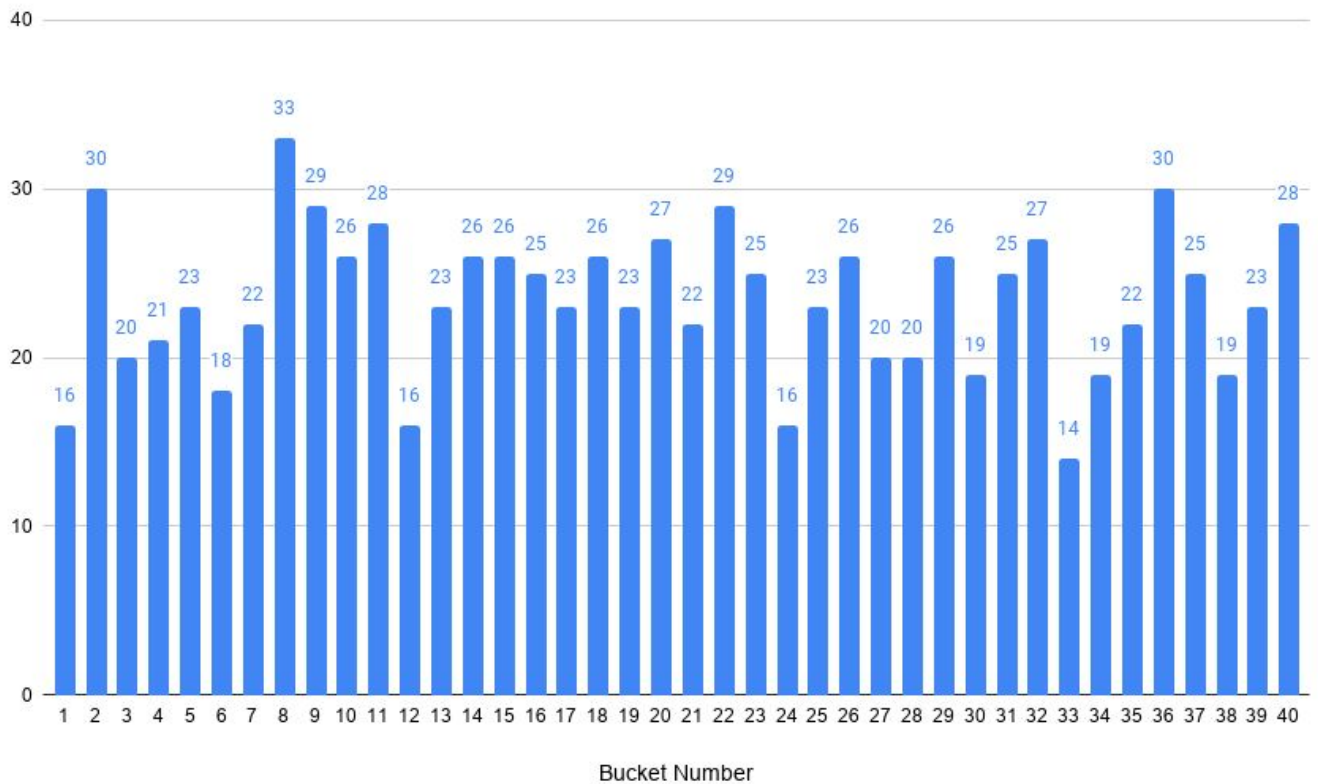
Average:26.828571

Deviation:3.6370019525874606

Minimum number of entries in buckets:19

Maximum number of entries in buckets:38

iii) For Hash Code 2 when bucket size 40



Total:939

Average:23.475

Deviation:4.233719102799127

Minimum number of entries in buckets:14

Maximum number of entries in buckets:33

When we compare hash codes 1 & 2 we can see that average entries per bucket is same for all hash codes when number of buckets are fixed. Also two hash codes were able to distribute keys through buckets in an almost uniform manner. when consider minimum, maximum hits and Standard Deviation of entries /bucket,

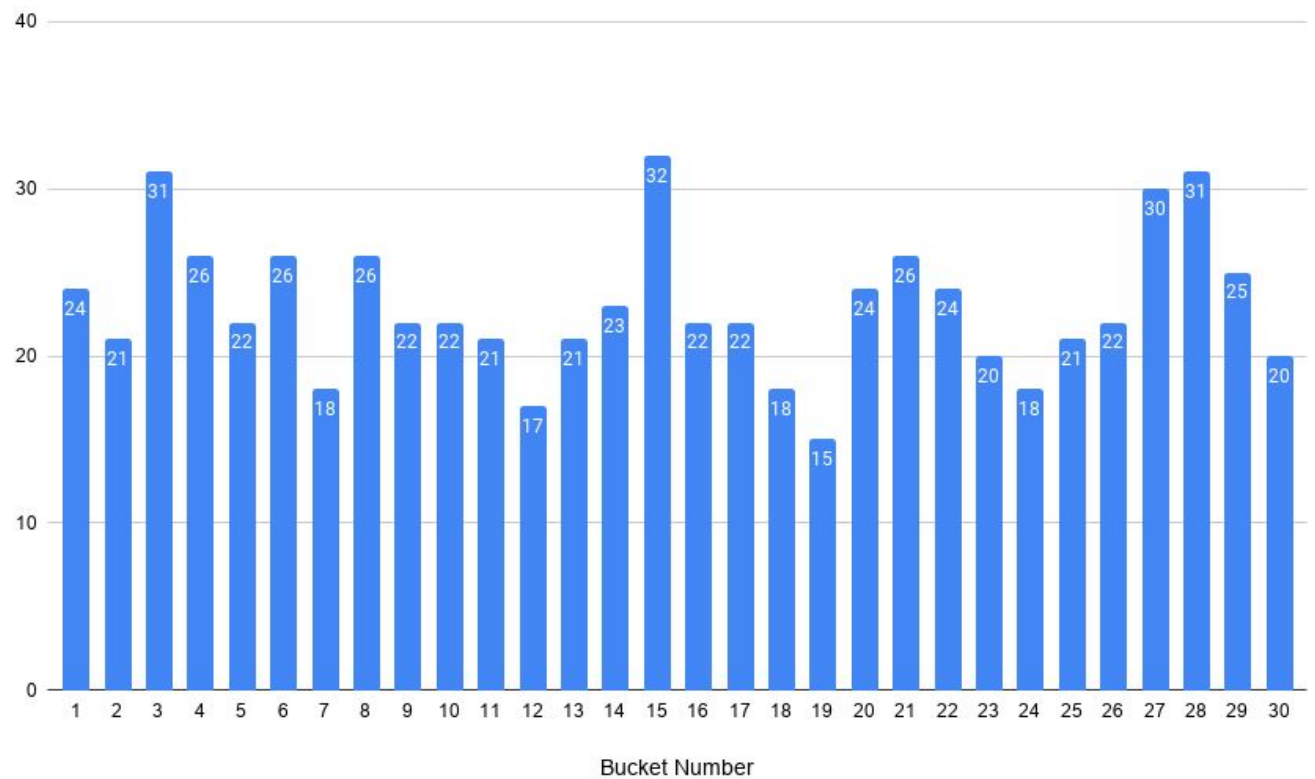
hash code 1 has better performances when number of buckets are 30.
hash code 2 has better performances when number of buckets are 35.
hash code 2 has better performances when number of buckets are 40.
Thus we can say that hash code 2 will be more suitable to expect a better performance of the hash table.

2) For sample-text2.txt file

a) Hash Code 1

```
private int hash(String key)
{
    int h = 0;
    for (int i = 0; i < key.length(); i++)
    {
        h = (31* h + key.charAt(i))%Table.length;
    }
    return (h%Table.length);
}
```


i) For Hash Code 1 when bucket size 30



Total:690

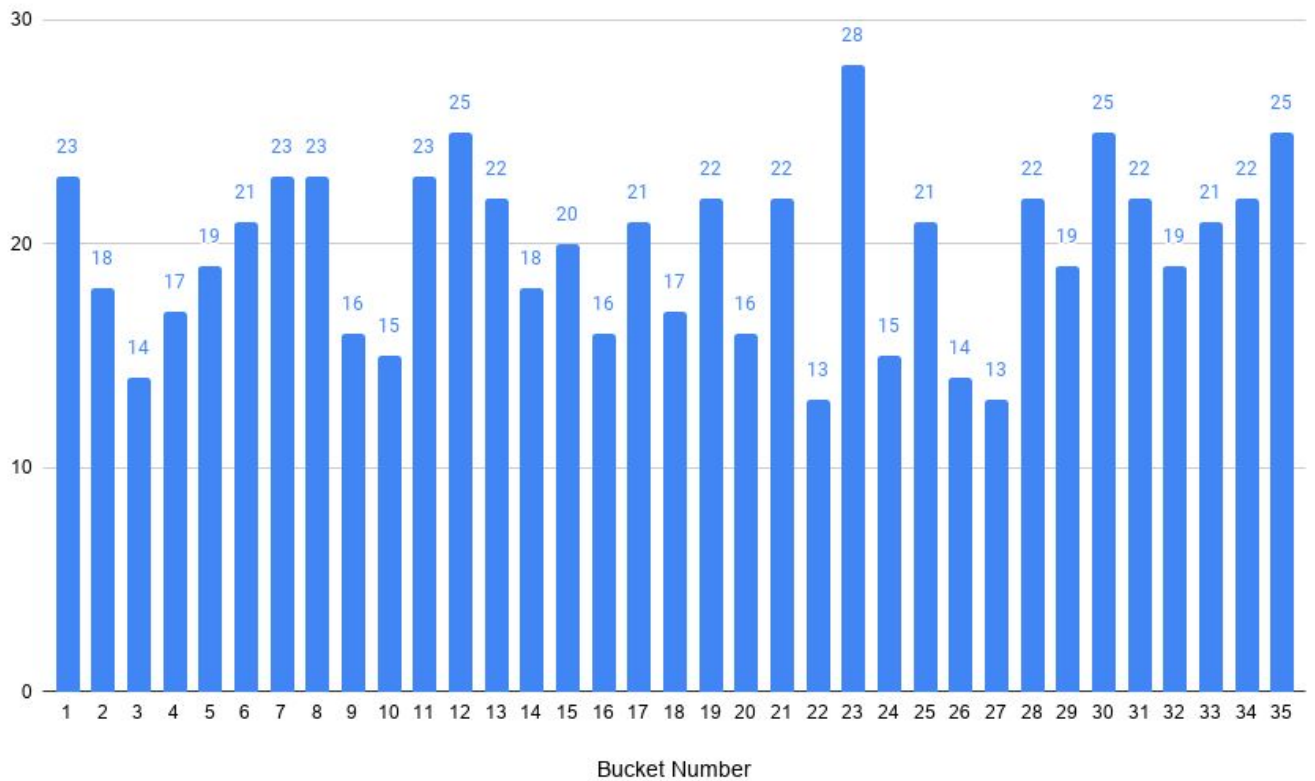
Average:23.0

Deviation:4.123105625617661

Minimum number of entries in buckets:15

Maximum number of entries in buckets:32

ii) For Hash Code 1 when bucket size 35



Total:690

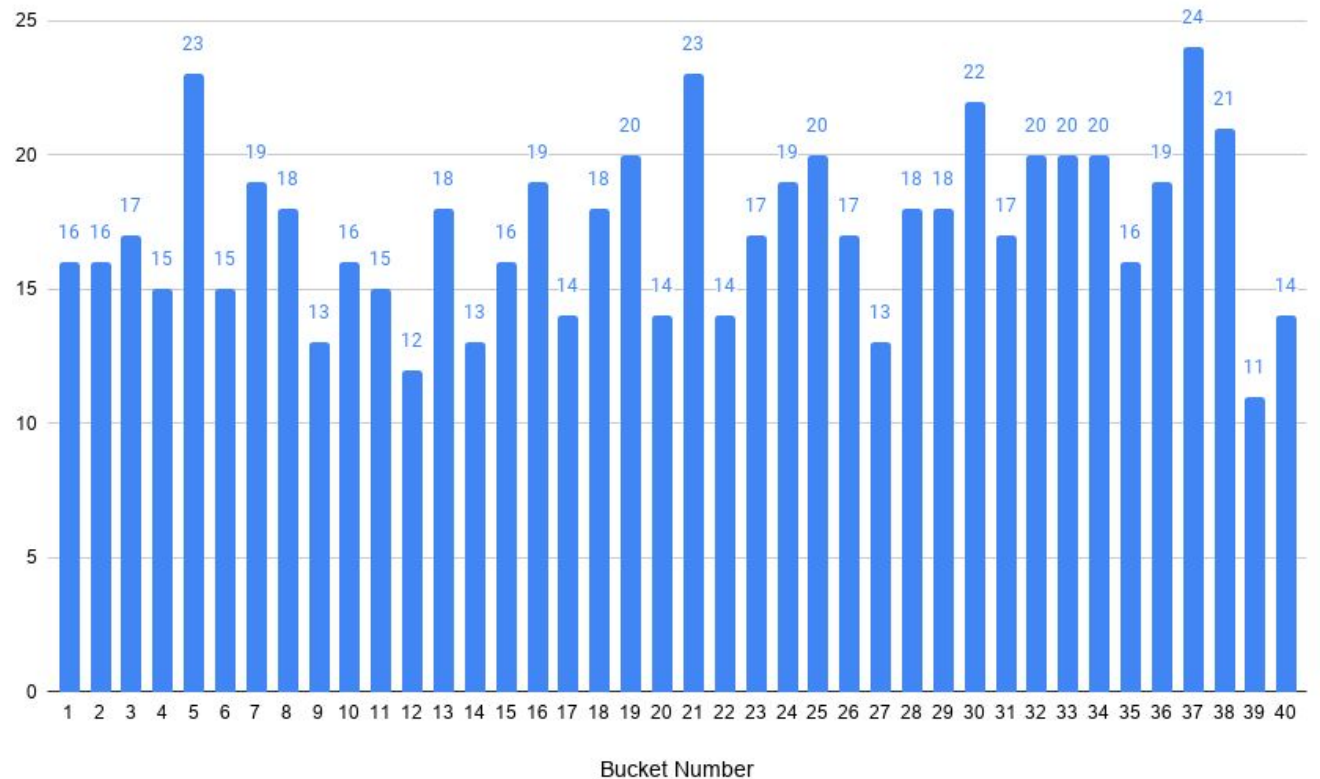
Average :19.714285

Deviation:3.653352551189284

Minimum number of entries in buckets:13

Maximum number of entries in buckets:28

iii) For Hash Code 1 when bucket size 40



Total:690

Average:17.25

Deviation:3.072051431861127

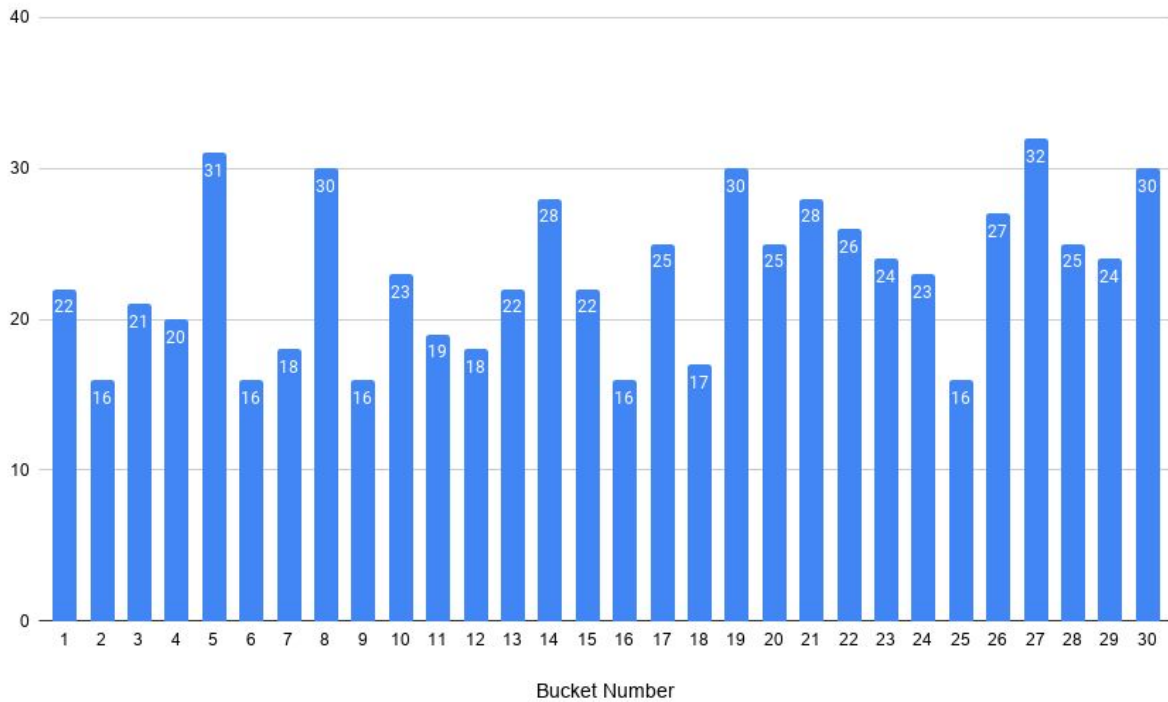
Minimum number of entries in buckets:11

Maximum number of entries in buckets:24

b) Hash Code 2

```
private int hash(String key)
{
    int h = 0;
    for (int i = 0; i < key.length(); i++)
    {
        h = (89* h + key.charAt(i))%Table.length;
    }
    return (h%Table.length);
}
```

i) For Hash Code 2 when bucket size 30



Total:690

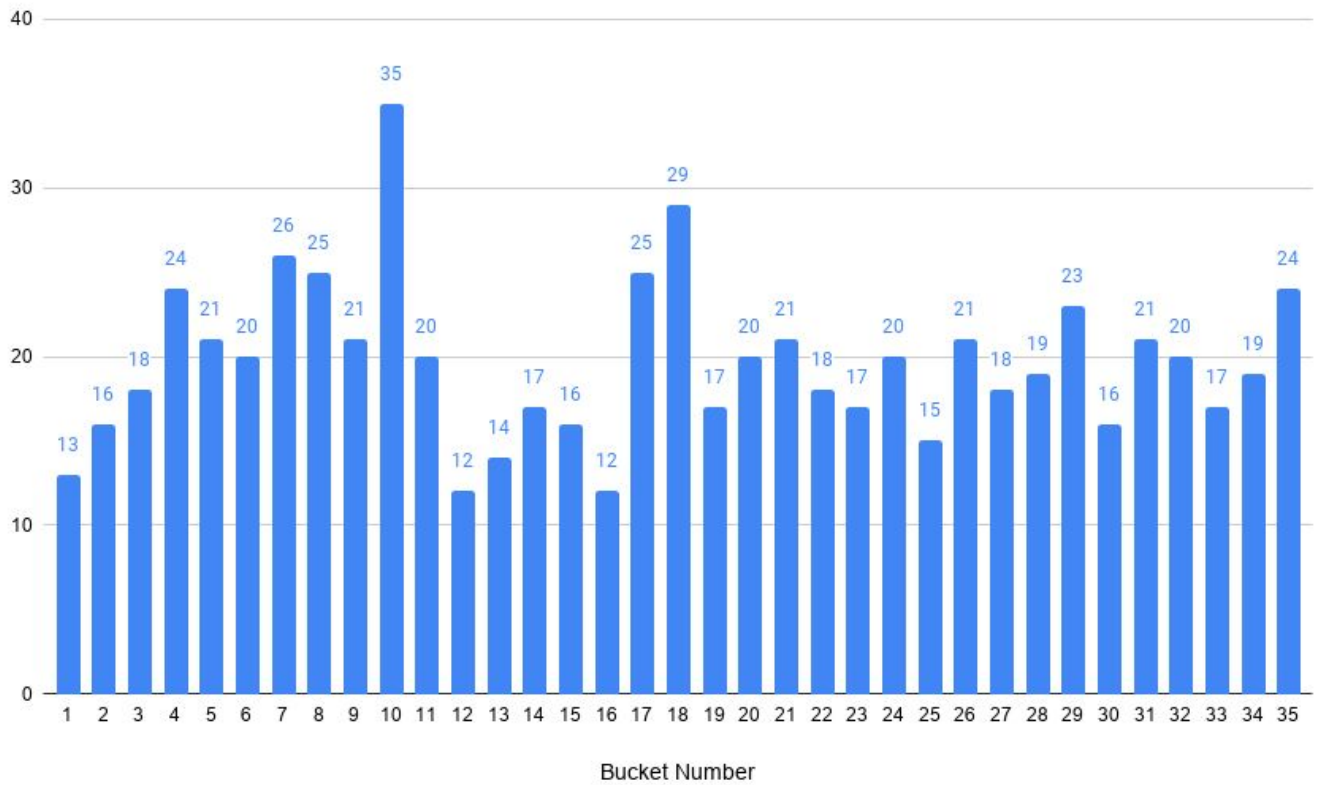
Average:23.0

Deviation:4.898979485566356

Minimum number of entries in buckets:16

Maximum number of entries in buckets:32

ii) For Hash Code 2 when bucket size 35



Total:690

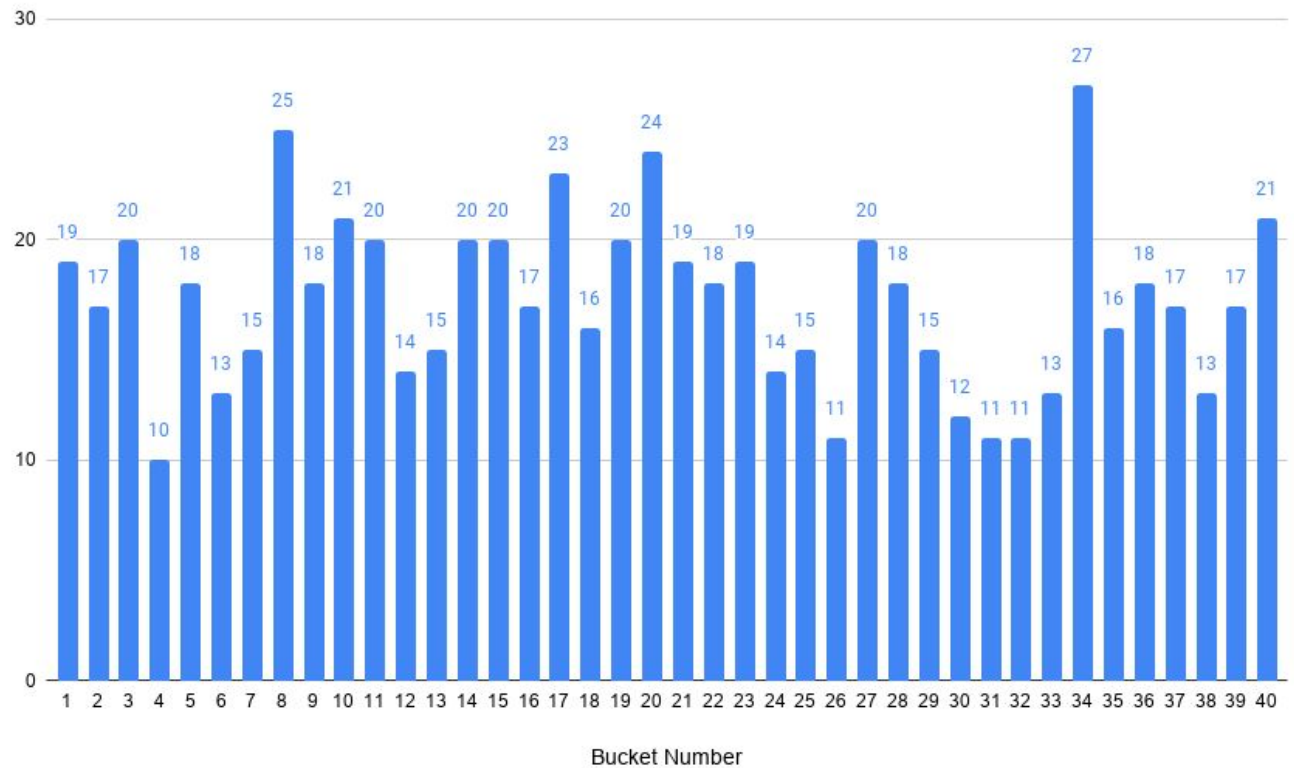
Average :19.714285

Deviation:4.620279738639344

Minimum number of entries in buckets:12

Maximum number of entries in buckets:35

iii) For Hash Code 2 when bucket size 40



Total:690

Average :17.25

Deviation:3.799671038392666

Minimum number of entries in buckets:10

Maximum number of entries in buckets:27

When the same hash function is used for different files it has given different results since the complexity of vocabulary used in 2 files is almost completely different therefore the domain of generated hash codes are also completely different. However despite the complexity of 2 files when an odd number is used as the multiplier in the hash function better uniformity has been occurred than when an even number is used. Also several odd

numbers has given the best uniformity above all other odd multipliers. Hash code 2 is an example for such situation.

Of course all above values changes for different bucket sizes. Here bucket sizes 25,30 and 35 are considered for all the situations for the easiness of comparing.