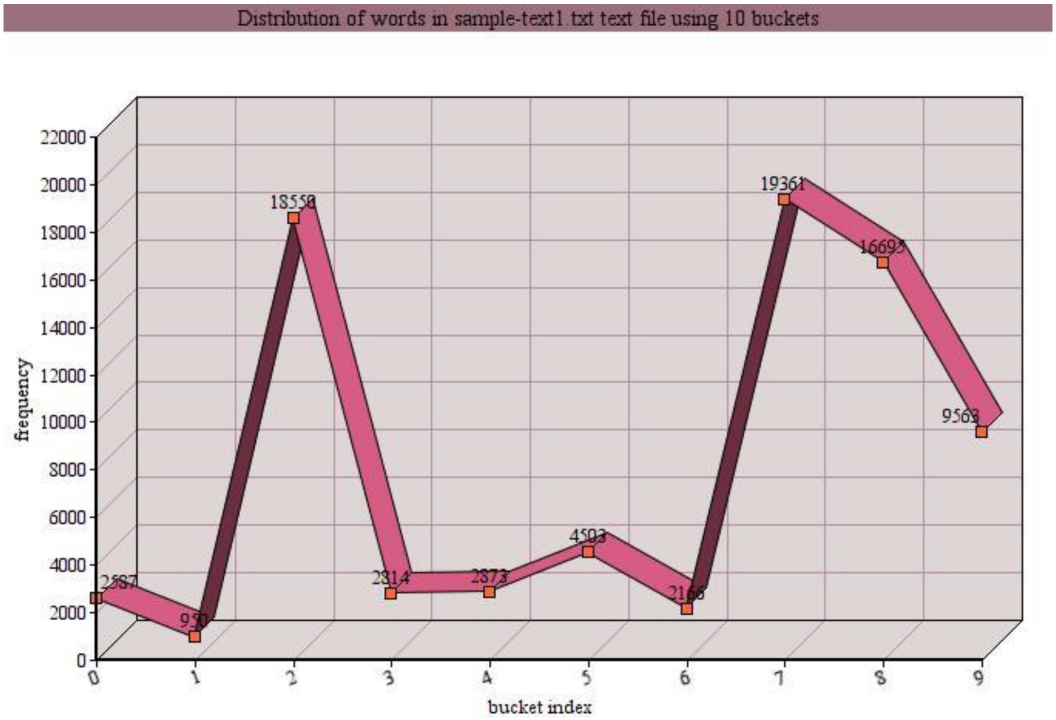
Chamara k.k.v

E/14/049

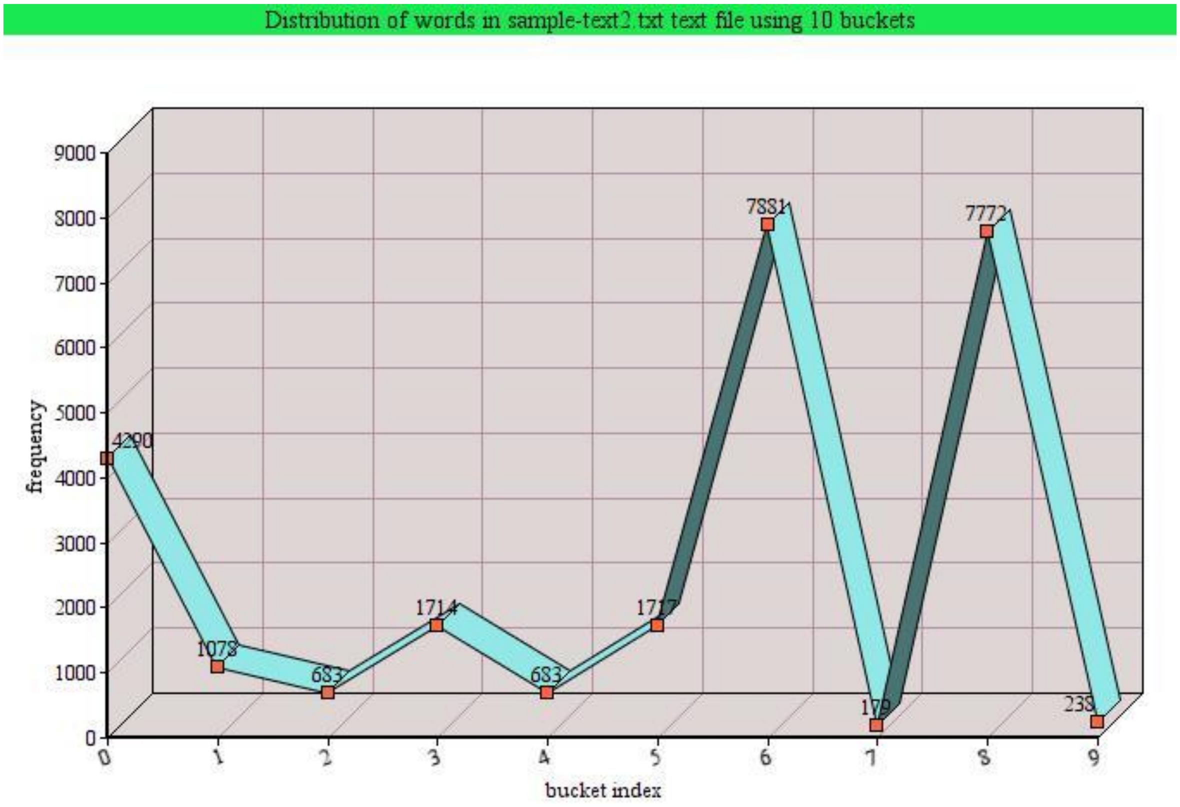
Using algorithm 1 and number of buckets as 10;

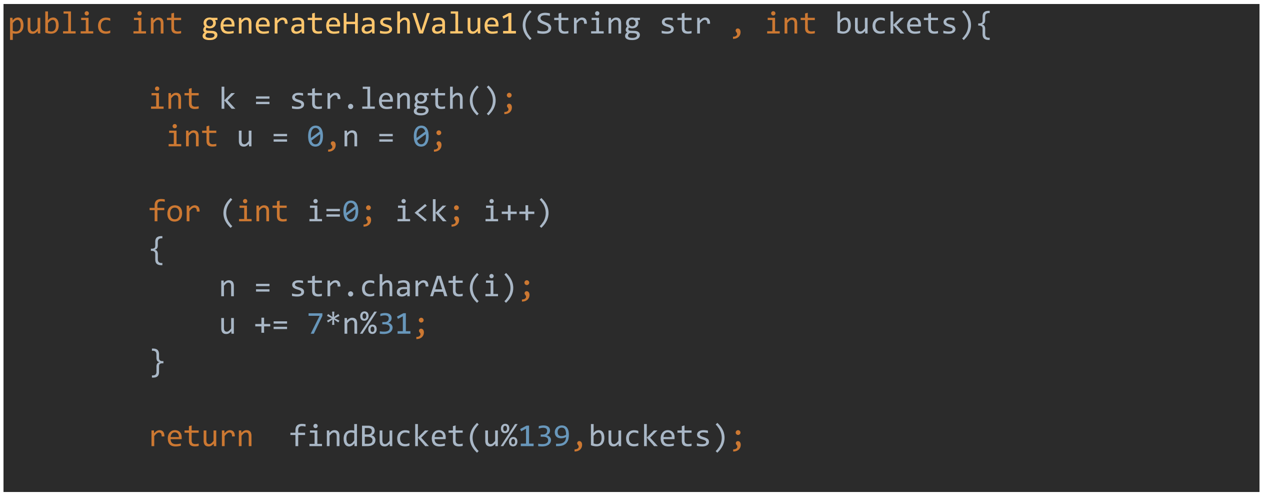
parameters	Textfile1.txt	Testfile2.txt
Average	8006.2	2623.8
Maximum	19361	78881
Minimum	950	179
Standard deviation	7430.185	2984.194



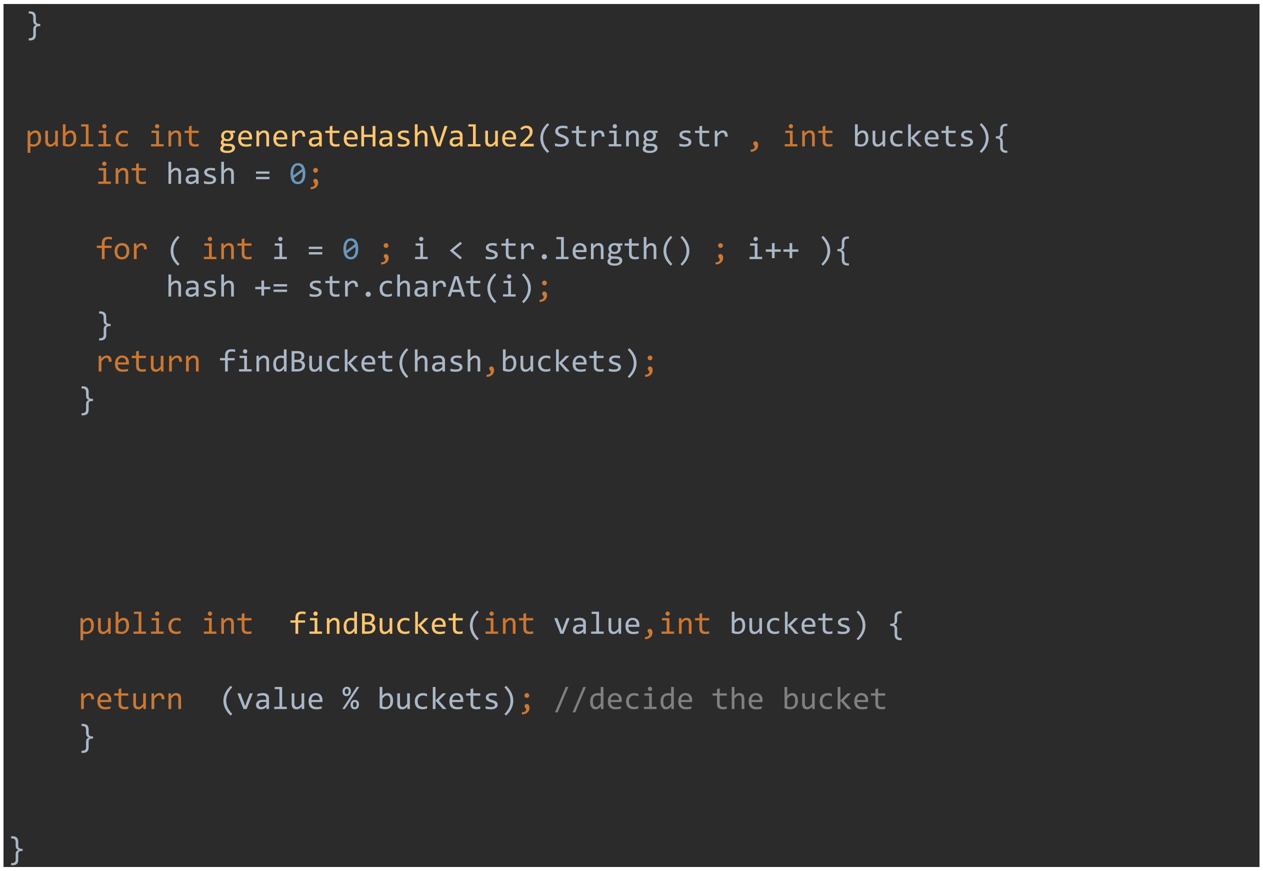
The hash functions that I have used are following. When, we take the text files 1 &2 they are quite different. For text file 1 when we take the number of buckets as 10, entries are not equality distributed. But compared with 10 buckets number 13 will give better result. Because 13 is prime number so that there are no factors in it. Therefore there will be high chance of occurring different bucket. Hence taking number of buckets as prime number more likely leads to better results.

As we can show in the table standard deviation will be low for the prime number of buckets. And average also high in frequency. Range is also high in prime number of buckets. Therefore taking prime as number of buckets will give better results.





```
public int generateHashValue1(String str , int buckets){

        int k = str.length();
         int u = 0,n = 0;

        for (int i=0; i<k; i++)
        {
            n = str.charAt(i);
            u += 7*n%31;
        }

        return  findBucket(u%139,buckets);
```

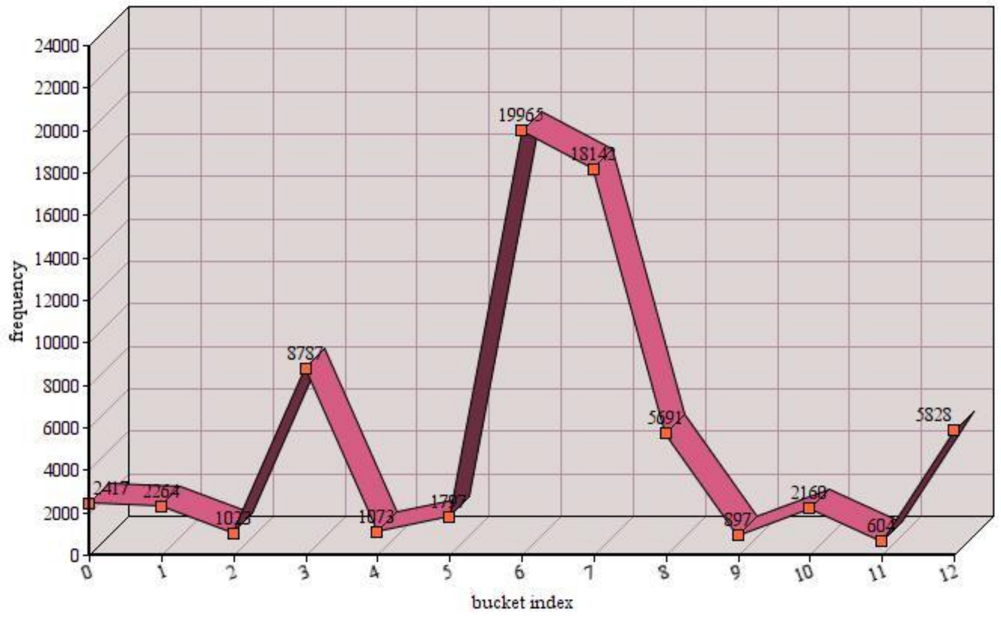


```
 }


 public int generateHashValue2(String str , int buckets){
     int hash = 0;

     for ( int i = 0 ; i < str.length() ; i++ ){
         hash += str.charAt(i);
     }
     return findBucket(hash,buckets);
    }




    public int  findBucket(int value,int buckets) {

    return  (value % buckets); //decide the bucket
    }


}
```

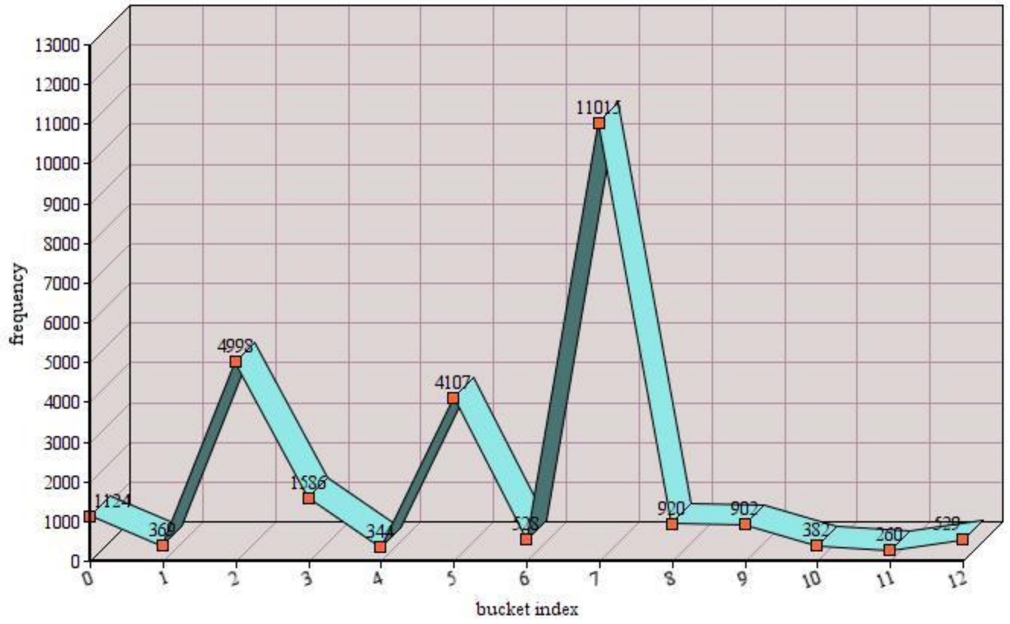
Using algorithm 1 and number of buckets as 13;

parameters	Textfile1.txt	Testfile2.txt
Average	5434.462	2081.846
Maximum	19965	11015
Minimum	604	260
Standard deviation	6516.605	3072.73

Distribution of words in sample-text1.txt text file using 13 buckets

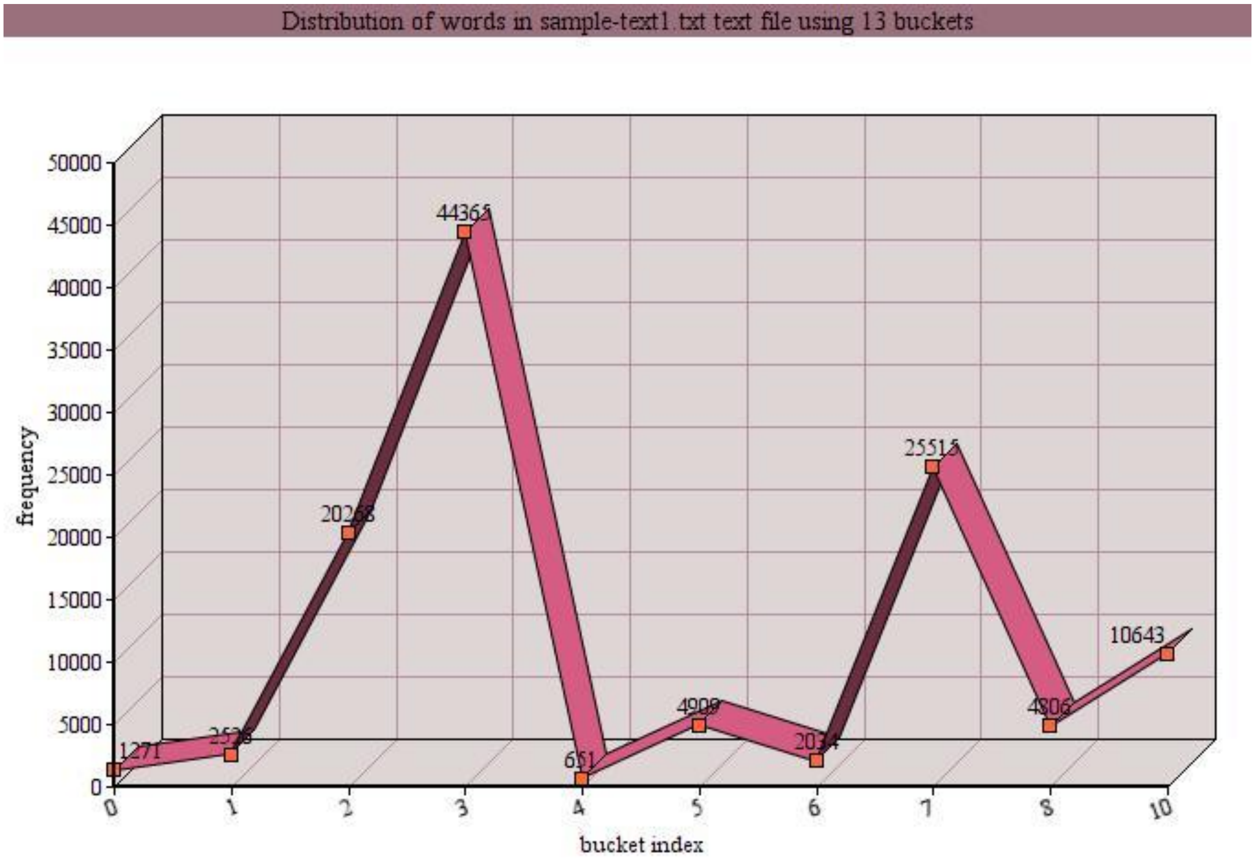


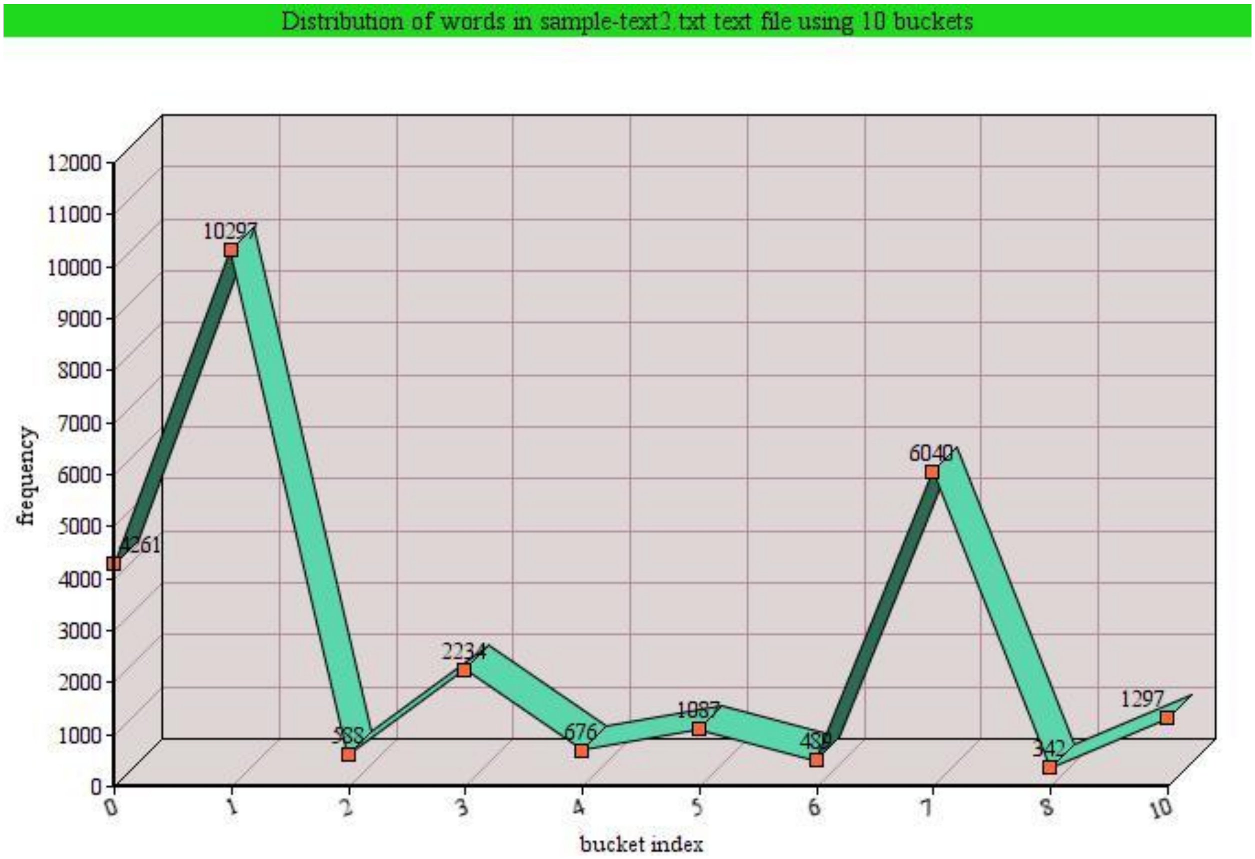
Distribution of words in sample-text2.txt text file using 13 buckets



Using algorithm 2 and number of buckets as 10 ;

parameters	Textfile1.txt	Testfile2.txt
Average	11698.8	2731.1
Maximum	44365	10297
Minimum	651	342
Standard deviation	14276.89	3250.473

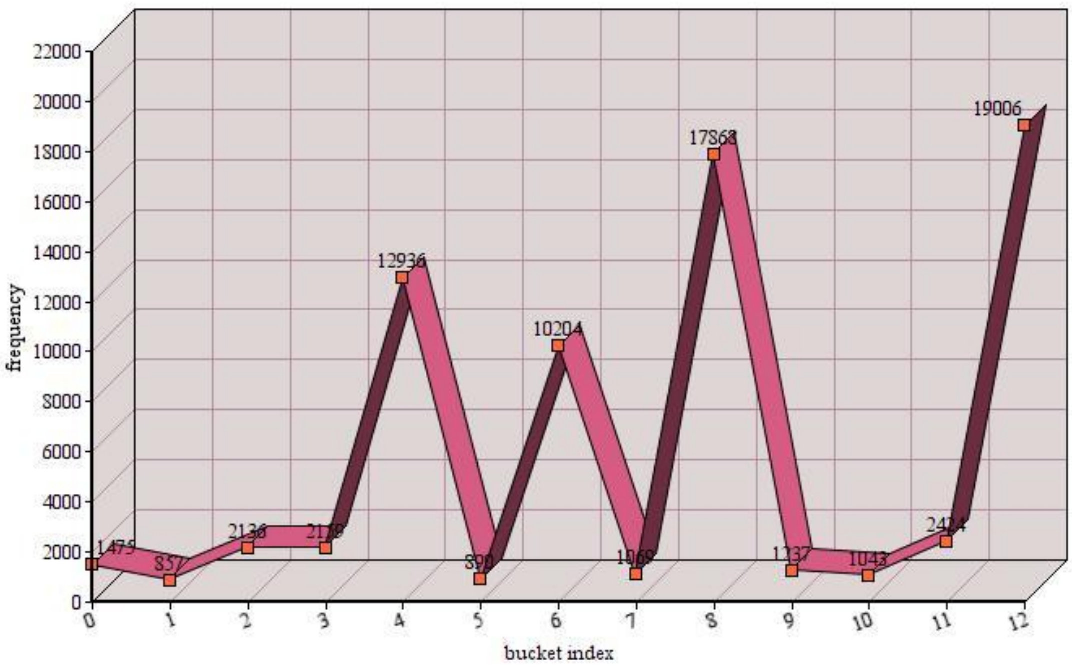




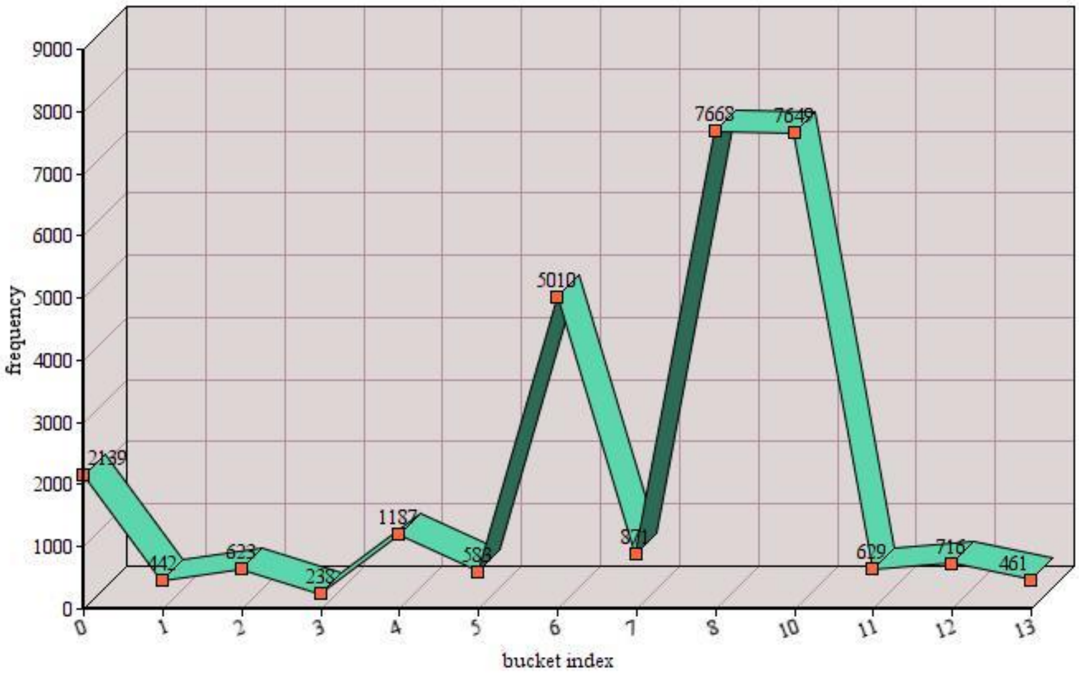
Using algorithm 2 and number of buckets as 13 ;

parameters	Textfile1.txt	Testfile2.txt
Average	5638.769	2170.462
Maximum	19006	7668
Minimum	857	238
Standard deviation	6838.453	2738.728

Distribution of words in sample-text1.txt text file using 13 buckets

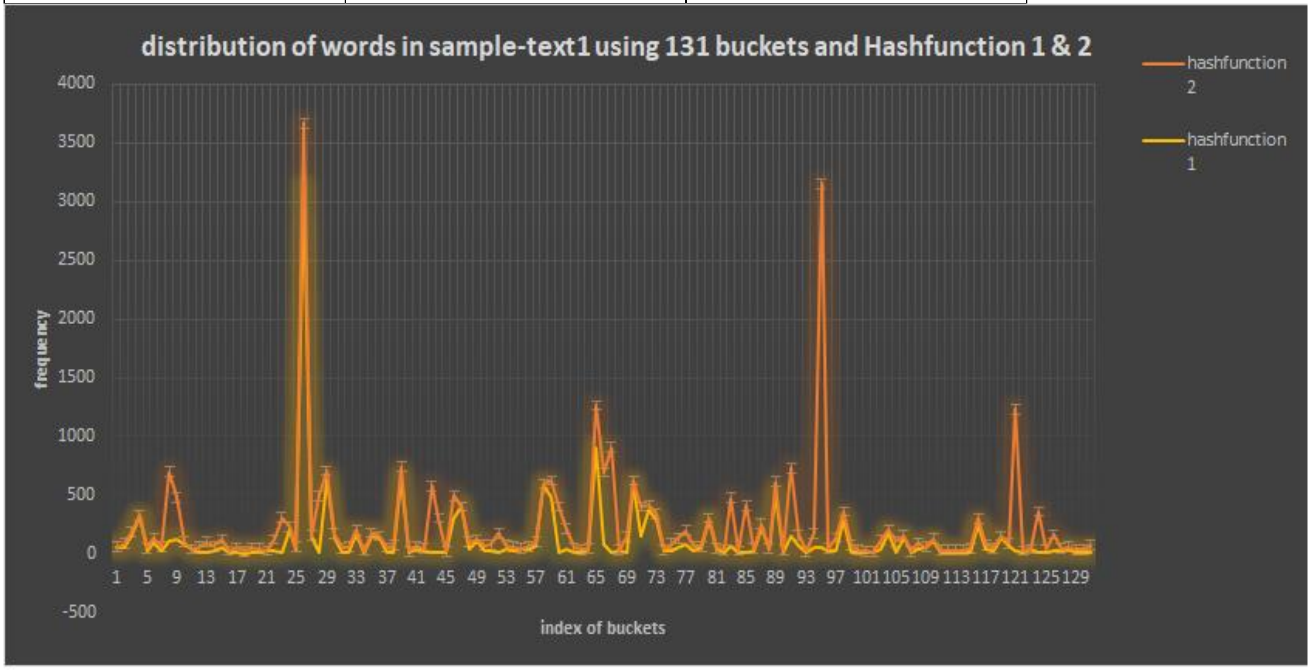


Distribution of words in sample-text2.txt text file using 13 buckets



<u>hashfunction 1 and 2 with number of buckets as 131 for file sample-text1 ;</u>

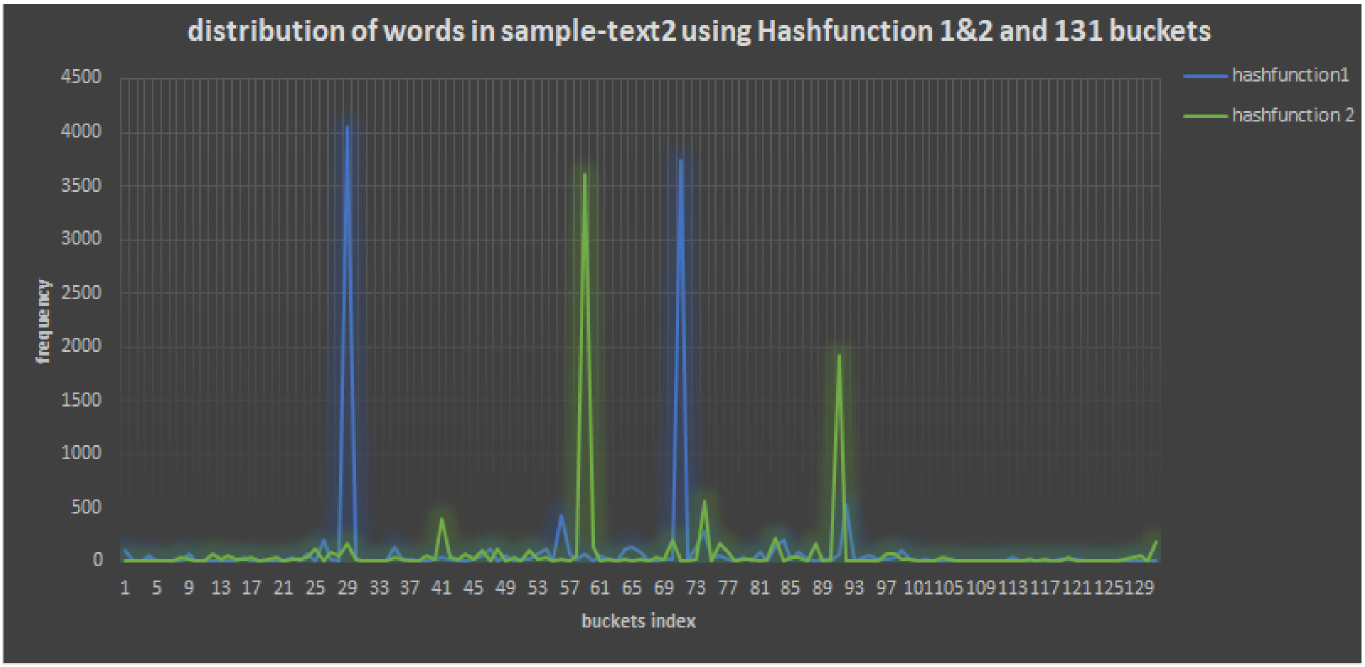
parameters	Hashfunction 1	Hashfunction 2
Average	115.2214	123.9237
Maximum	3131	3099
Minimum	2	3
Standard deviation	307.4164	318.6244



If we increase the buckets then the distribution more likely be equally distribute. On the other hand memory is sacrificed.   Considering average and the standard deviation hash function 1 suitable for the sample-text1 file and hash function 2 suitable for sample-text2 file.

hashfunction 1 and 2 with number of buckets as 131 for file sample-text2;

parameters	Hash function 1	Hash function 2
Average	98.36641	80.58779
Maximum	4045	3607
Minimum	0	0
Standard deviation	481.636	359.5194


distribution of words in sample-text2 using Hashfunction 1&2 and 131 buckets
hashfunction1
hashfunction 2
frequency
buckets index