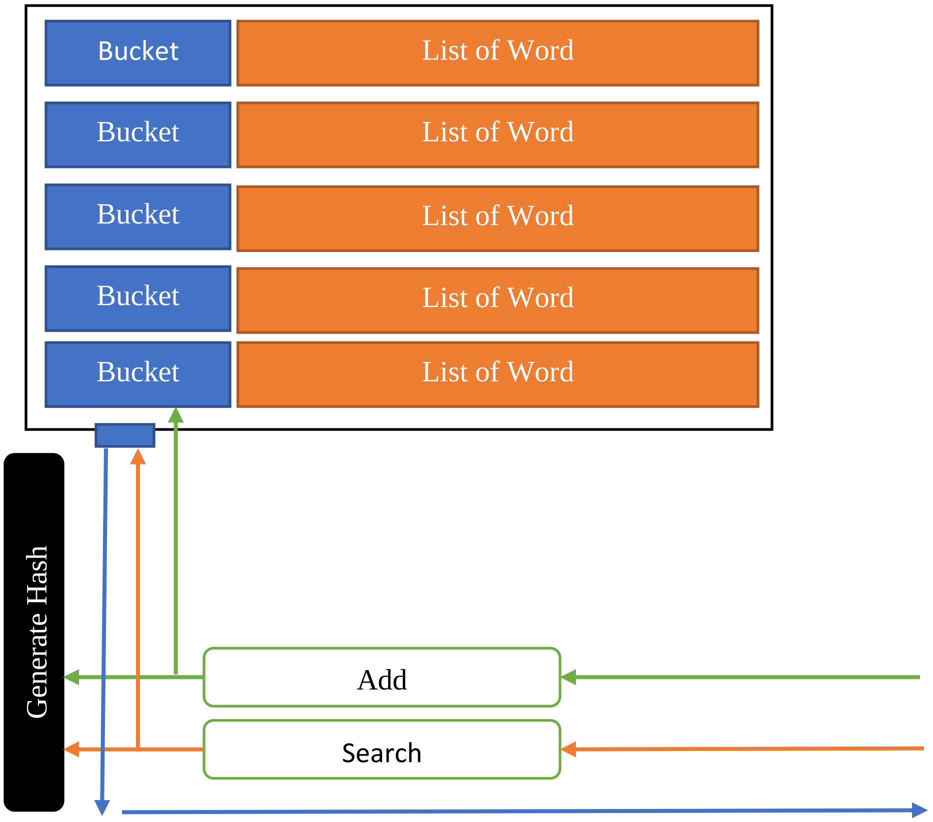
Hash Class Structure:



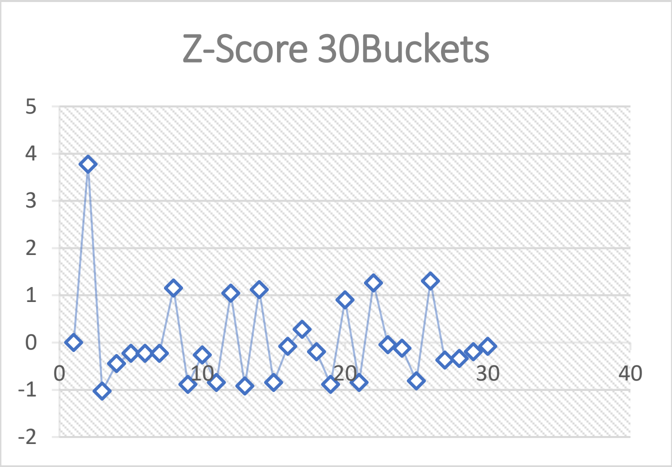
This has two main functions and one for add and another for get value, set of words are kept in Bucket Array. Size can be defined at initializing. Hash code for each word is generated by Hash Generator method. That word stores according to that hash. To search a word search method only search on hash bucket and return count of the word.

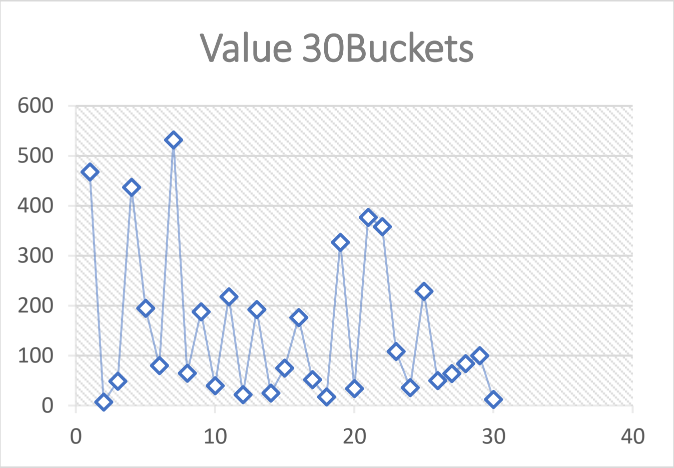
Hash code can be generated by several methods, and each try has deferent bucket lengths, means, Zs. In good concept has low standard deviation for many bucket sizes.

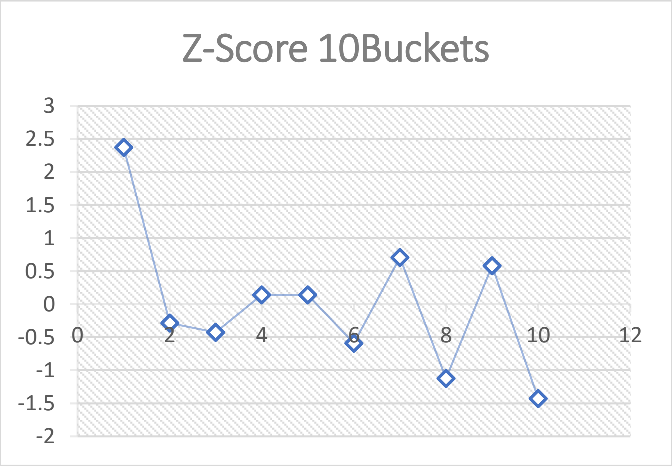
Concept 1:

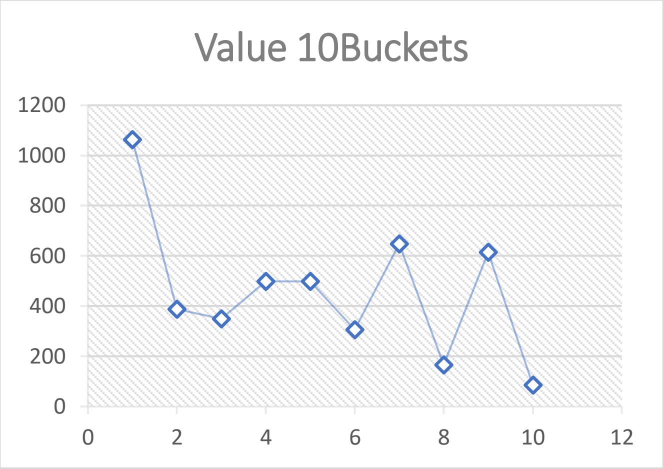
```
public int genarateHash(String word) {
    int hash = 0;
    for (int i = 0; i < word.length(); i++) {
        hash += word.charAt(i);
    }

    return word.length() * hash % this.bucketSize;
}
```

Total Word Count:4617

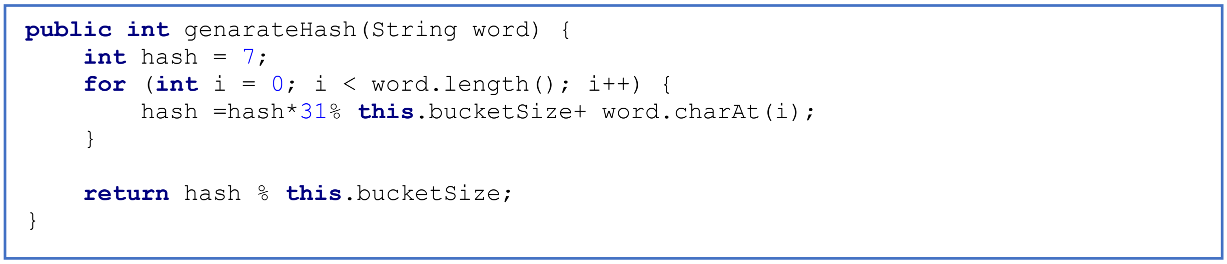

Z-Score 30Buckets


Value 30Buckets

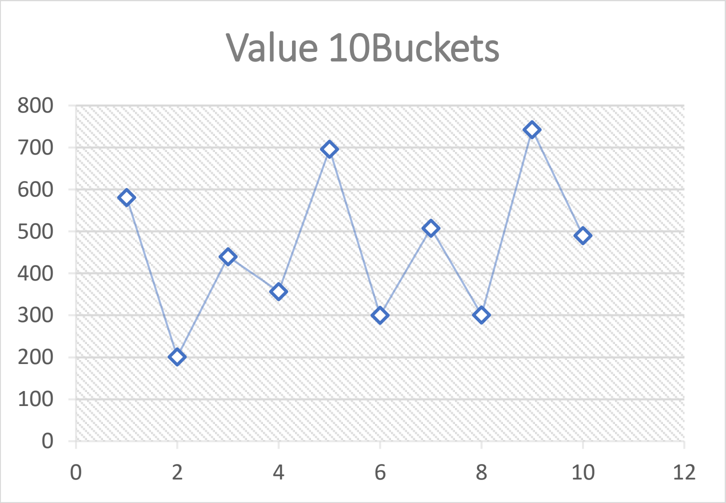

Z-Score 10Buckets

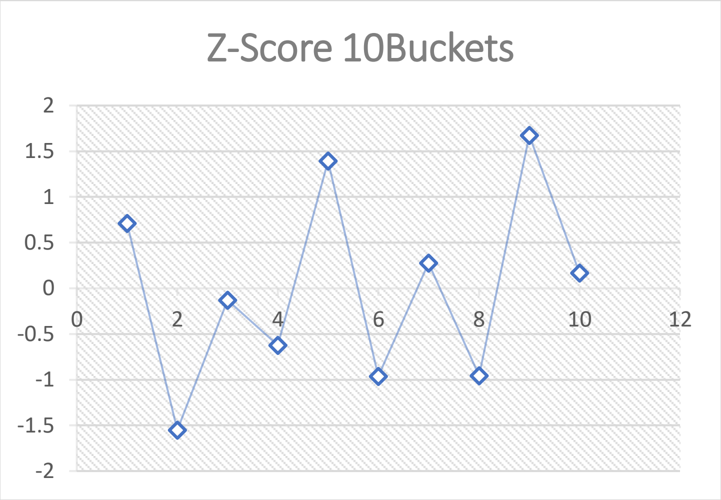

Value 10Buckets

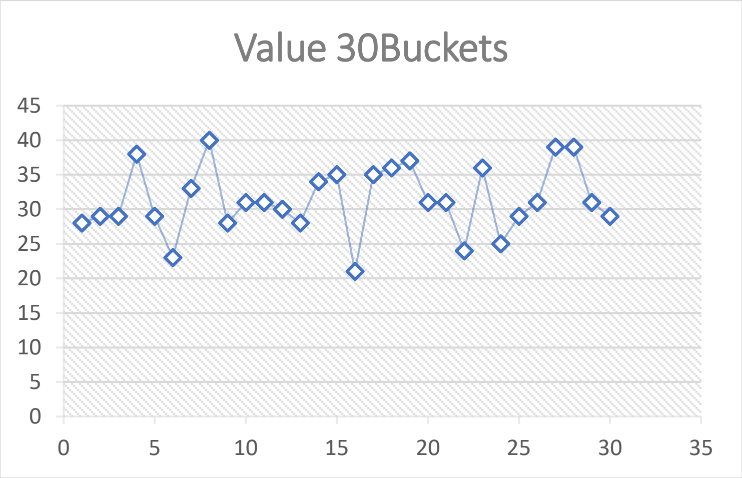
E/14/349 Thilakarathna BRM CO322 Lab-Hash Table

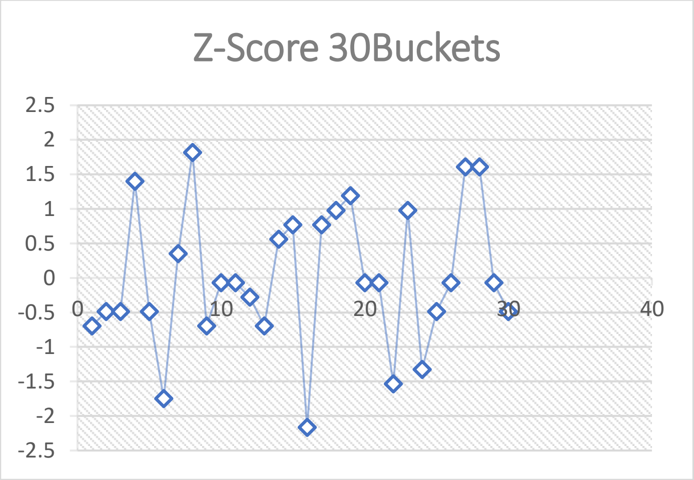


```
public int genarateHash(String word) {
    int hash = 7;
    for (int i = 0; i < word.length(); i++) {
        hash =hash*31% this.bucketSize+ word.charAt(i);
    }

    return hash % this.bucketSize;
}
```

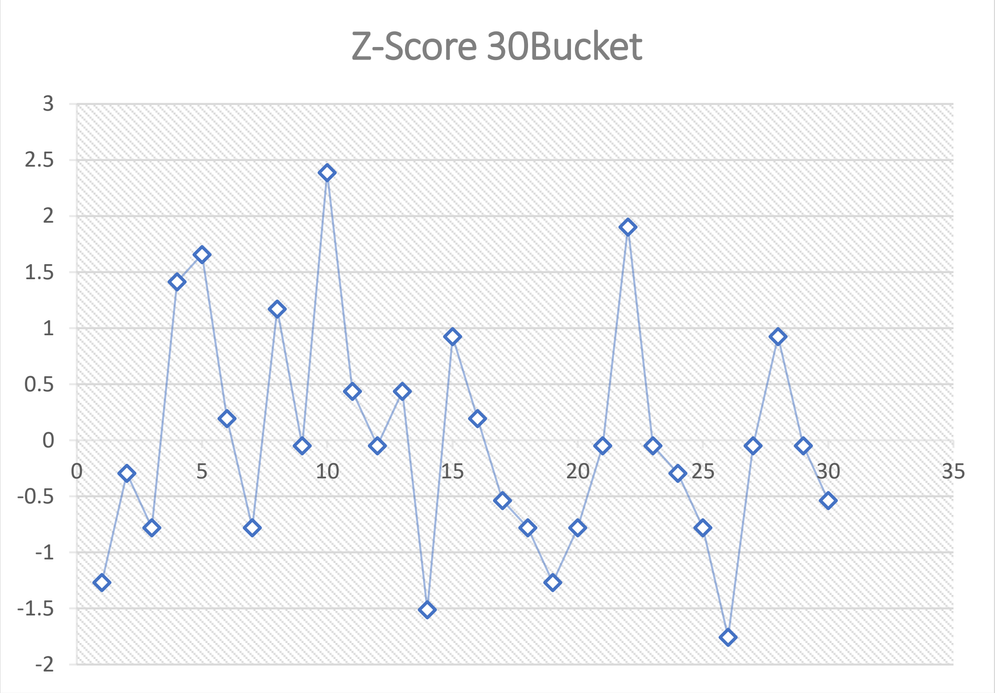


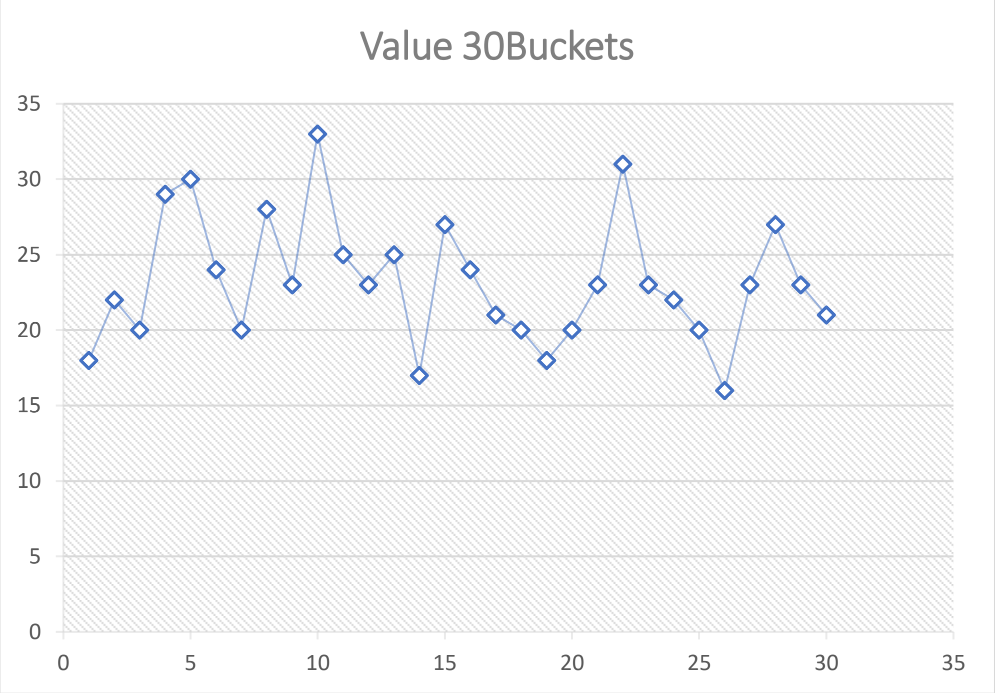


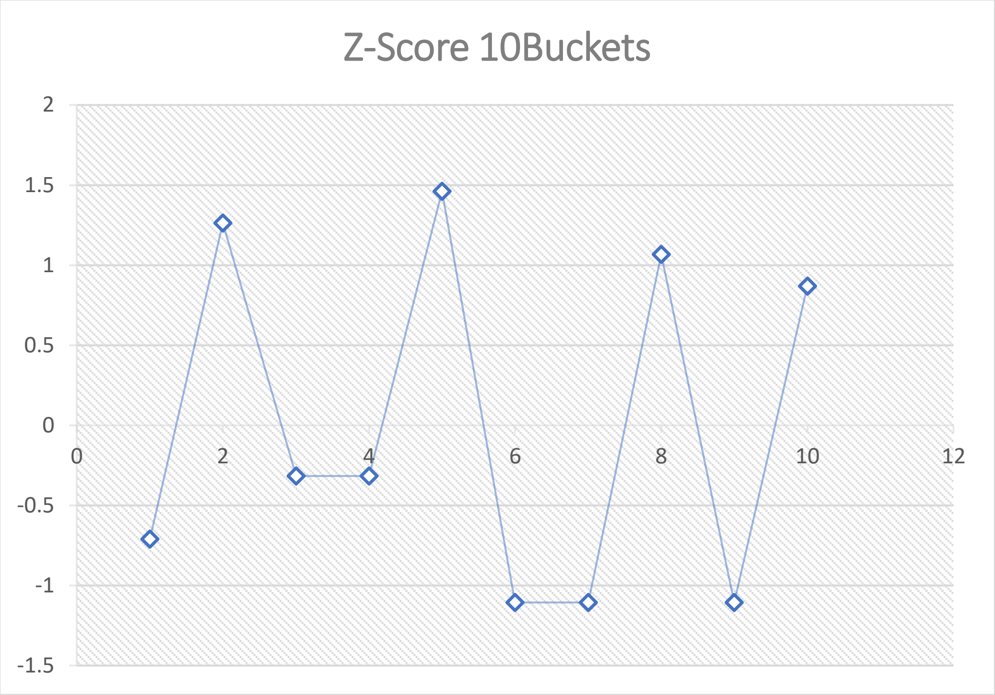


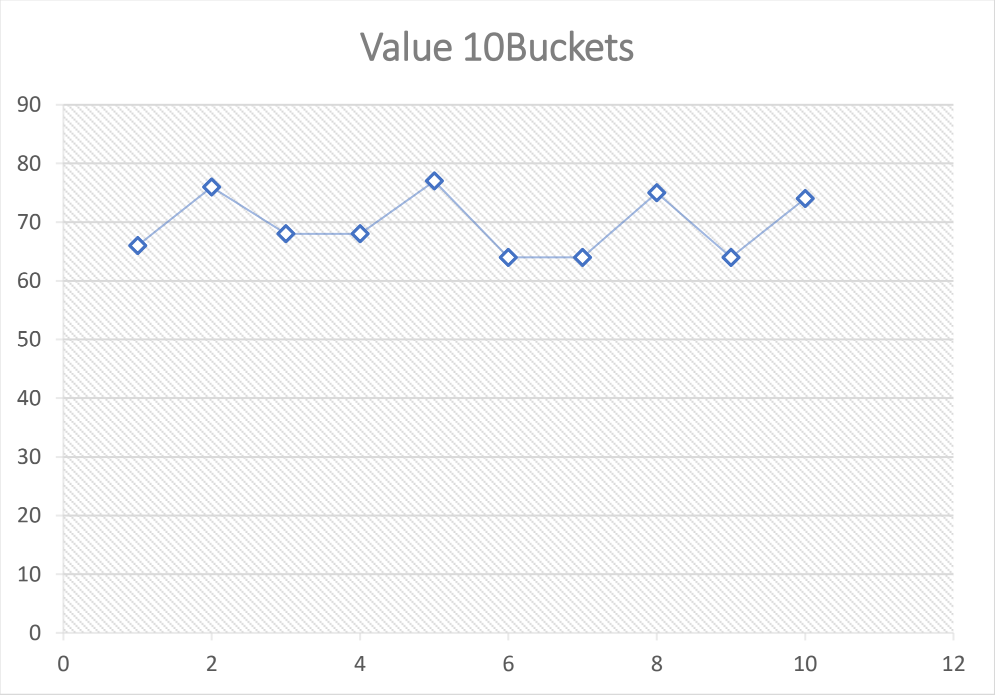


We can 2^{nd} method Is better than first one. Let's look at 2^{nd} text for this 3nd method.









Seems it is okay to 2nd text and it keep good bucket sizes.