CO322-Lab03

Report

K.A.H.P.Thilakarathna

E/15/362

Implemented hash functions

Hash_function1

Initialize key=0

Input word

key = s.length()_modulus_buckets

Hash function2

initialize total=0, i=0 and key=0

Input word

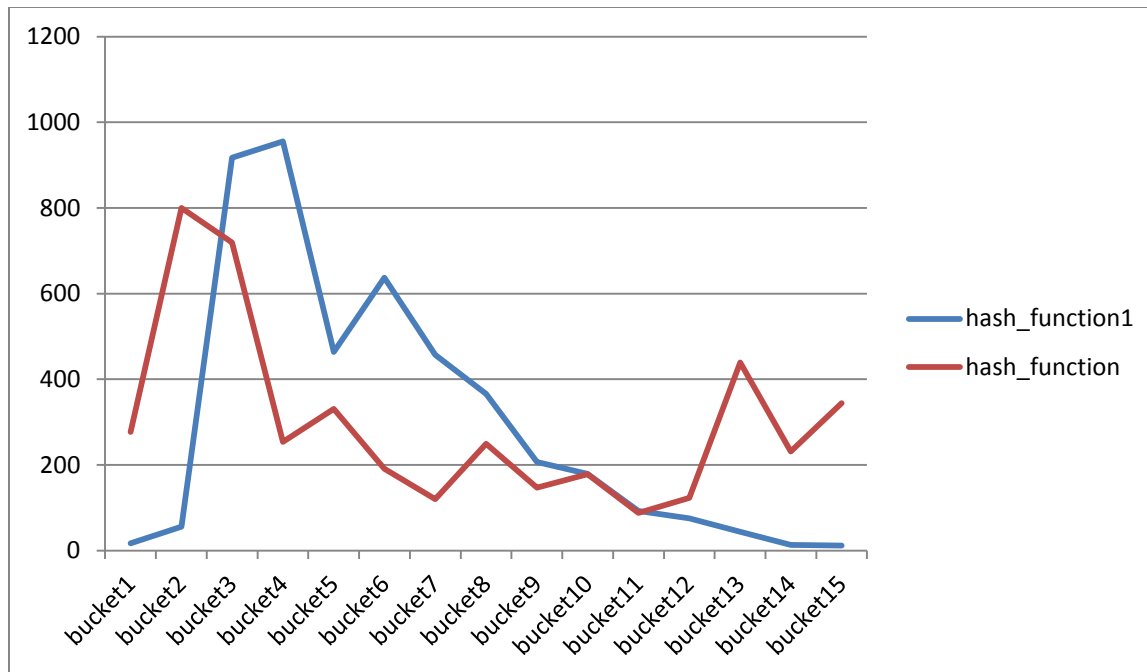separate_letters_as_characters_in_word

For i=0:word_length:

total =total + character_at_i_position_in_word

End for

key = (total/i)_modulus_buckets;

Distribution of **bucket sizes** for **sample-text1.txt** when different hash functions are used with **15 buckets**

In the above graph we can see the distribution of bucket sizes for different hash functions when 15 buckets are used. When *hash_function2* is used, we can see a comparatively more evenly distributed graph as the entries are distributed among the buckets more evenly .

	Hash_fucntion1	Hash_fucntion2
Bucket1	17	277
Bucket2	56	800
Bucket3	917	719
Bucket4	955	254
Bucket5	464	331
Bucket6	637	191
Bucket7	457	120
Bucket8	366	249
Bucket9	207	147
Bucket10	179	178
Bucket11	92	88
Bucket12	75	123
Bucket13	44	439
Bucket14	13	231
Bucket15	12	344

Distribution of **bucket sizes** for **sample-text1.txt** when different hash functions are used with **20 buckets**
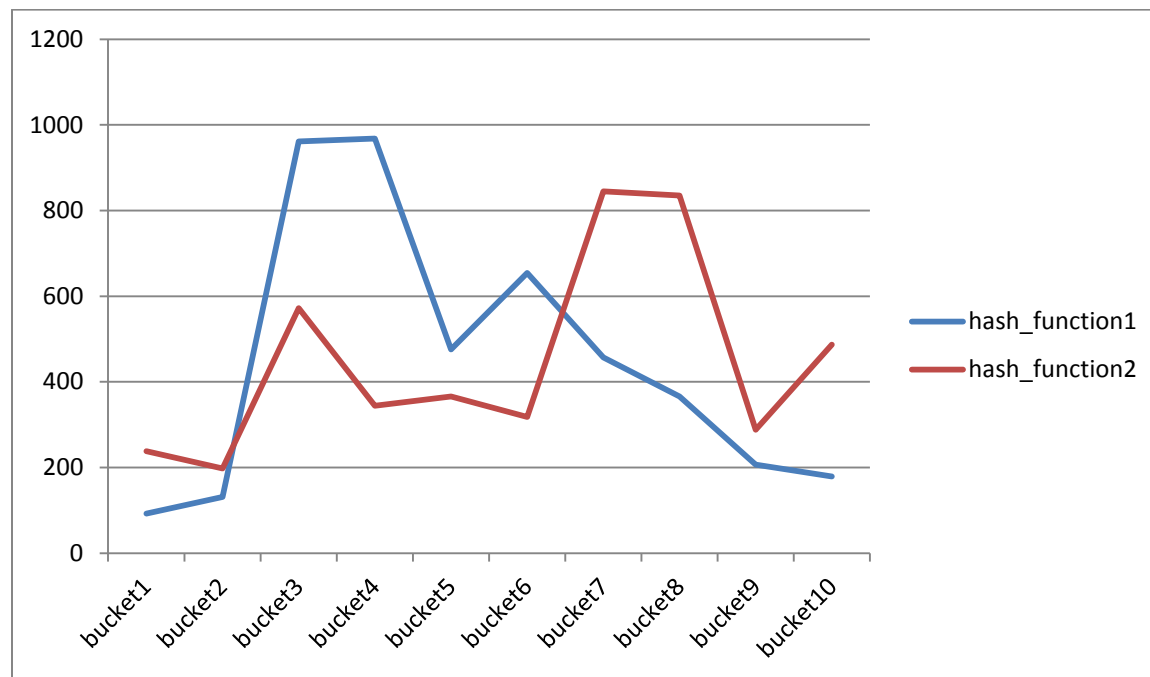
Here the 20 buckets have been used. If you carefully look at the above graph you can see both functions distribution has been changed when the bucket size is changed from 15 to 20.

And also for both functions using 20 buckets gives better hash tables than when 15 buckets are used. Thus, we can say that the **bucket size also matters** when obtaining a better hash table.

Given below is the **maximum and minimum number of entries** among buckets for different hash functions (for *sample-text1.txt*). Here we can see considering those two data, that **hash_function2 has a better range** which again shows that the distribution of entries among the buckets for hash_function2 are more even compared to hash_function1.

	Hash_fucntion1	Hash_fucntion2
maximum	955	800
minimum	12	88

You can see the corresponding graph below.

Distribution of **maximum and minimum number of entries** for **sample-text1.txt** when different hash functions are used with **15 buckets**
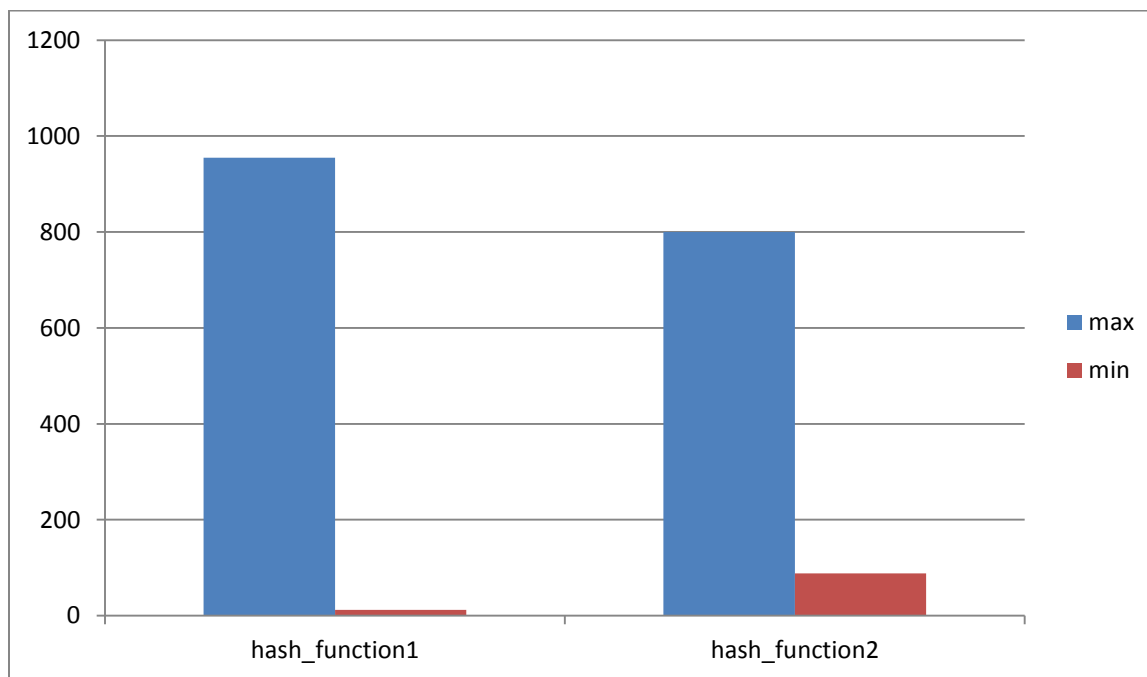
Variation of **Standard deviation** for **15 buckets** are shown below.

When **sample-text1.txt** is used,

	Hash_fucntion1	Hash_fucntion2
Standard deviation	312.71194	202.75233

Standard deviation for hash_function2 is low compared to hash_function1 which shows that the hash_function2 has more evenly distributed hash table (for sample-text1.txt).

****

Now let's see all the above comparisons and details when **sample-text2.txt** is used.

In the following graph we can see the distribution of entries among buckets for different hash functions when sample-text2.txt is used. Here, if you see carefully the both graph and the data table you will see that the hash_function2 has more evenly distributed buckets than the hash_function1 .

Distribution of **bucket sizes** for **sample-text2.txt** when different hash functions are used with **15 buckets**
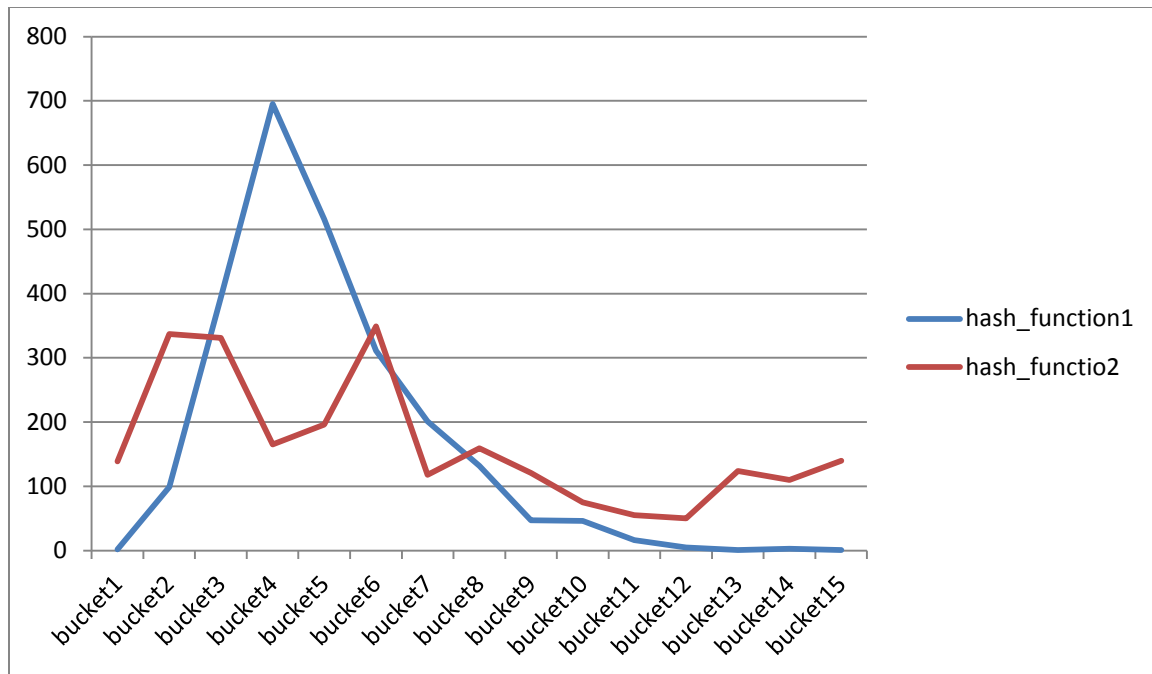
	Hash_function1	Hash_function2
Bucket1	2	139
Bucket2	99	337
Bucket3	394	331
Bucket4	695	165
Bucket5	516	196
Bucket6	311	349
Bucket7	201	118
Bucket8	132	159
Bucket9	47	121
Bucket10	46	75
Bucket11	16	55
Bucket12	5	50
Bucket13	10	124
Bucket14	3	110
Bucket15	10	140

Distribution of **maximum and minimum number of entries** for **sample-text2.txt** when different hash functions are used with **15 buckets**
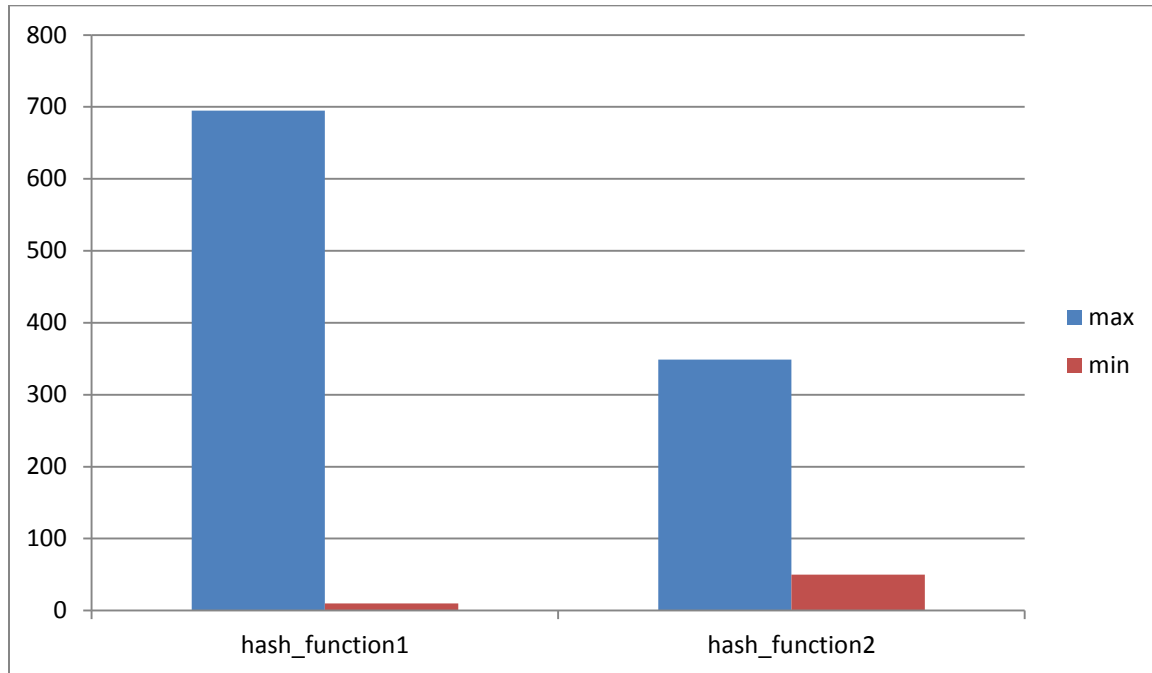
	Hash_fucntion1	Hash_fucntion2
maximum	695	349
minimum	10	50

Range in hash_function2 is much better than the range in the hash_function1 which again shows that the hash_fucntion2 has a more evenly distributed hash table.

Variation of **Standard deviation** are shown below.

	Hash_function1	Hash_fucntion2
Standard deviation	210.88506	95.06756

Again, above data shows that hash_function2 has provided a better distribution in the hash table than the hash_function1.

- When considering all the above details given, we can see that when the **text file is different, the distribution differs** for both hash functions as none of those hash functions depend on the text file.