E/16/388

Weerasundara W.M.T.M.P.B.

CO 322 Data Structures and Algorithms

Lab 03 - Hash Tables

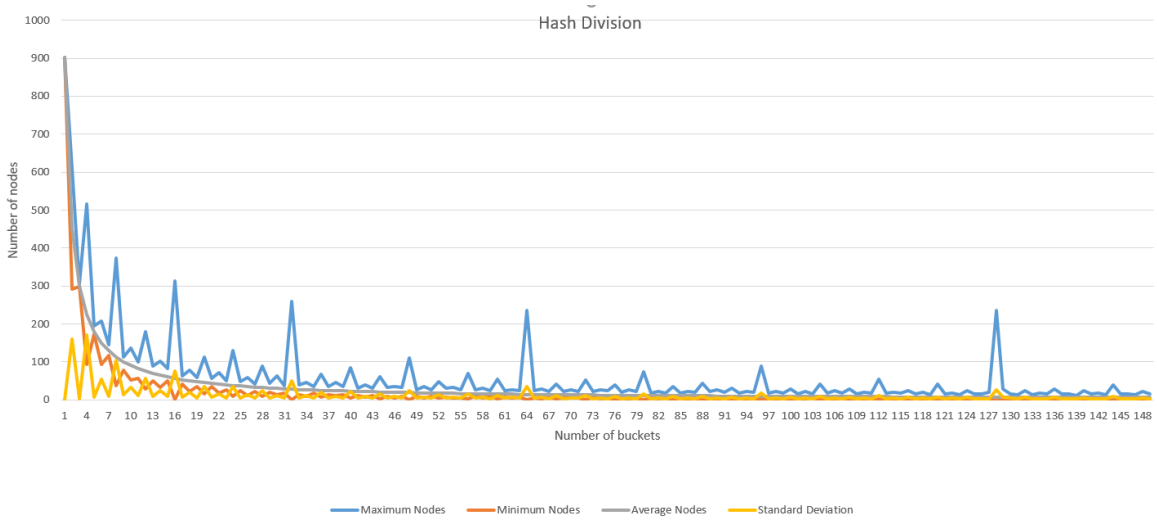
Let's consider about the hash division and multiplication methods. In here words are stored in a hashtable using the above 2 methods. Let's consider this for 2 samples under different number of buckets.

Distribution for sample 1

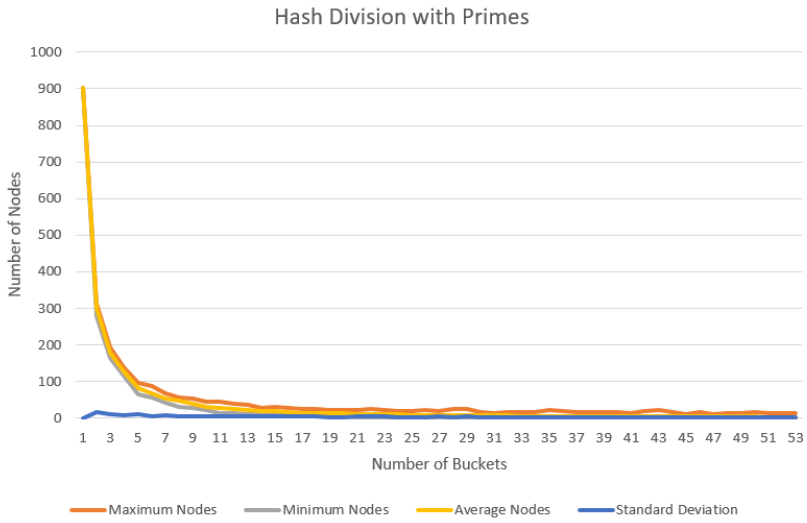
Consider the Divison method for hashing 1st.

h(k) = k mod m; k is the key and m is the number of nodes m > 0 and m ϵ Z

For this method consider the plots of maximum and minimum number of nodes in a bucket, average number of nodes in a bucket and standard deviation of nodes in a bucket for the sample 1 text file.



As we can see there are lots of deviations in all 4 data sets so this plot is incosistent. Specially around powers of 2, there are local peaks and it gives higher standard deviation values. Standard deviation is the best parameter of the considered 4 parameters. Hence chosing this method with this number of buckets is no very efficient. When choosing powers of 2 as the number of buckets, hashing does not depends on the all bits in the key. Therefore hashin is inefficient. Reason is when taking mod from power of 2, we directly only select the least significant bits of the key and rest of the bits are neglected. Therefore by avoiding powers of 2 we can improve this method. Also if we use prime numbers as number of buckets, since primes does not divisible, we can further improve our hashing method.



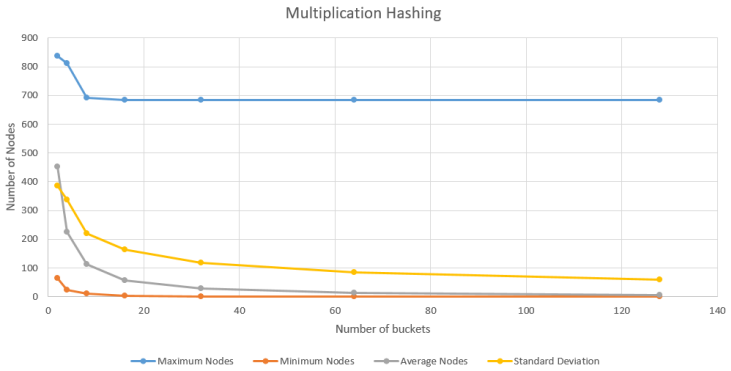
In this plot we can see the curves are more smooth since we use prime numbers as buckets. Therefore the hash function is much efficient than the previous time.

Also when searchin for an element we do not mind 3 unsuccessful searches in average. Therefore at minimum we can have 4 elements in a linked list. With 3 unsuccesful searches we can get the correct element with the 4^{th} search succesfully.

Number of Buckets	Maximum Nodes	Minimum Nodes	Average Nodes	Standard Deviation
211	12	0	4.274881517	2.915928532
223	14	0	4.044843049	2.859608592
227	13	0	3.973568282	2.905138551
229	18	0	3.938864629	3.005921452
233	14	0	3.871244635	2.846681178
239	13	0	3.774058577	2.680940916
241	15	0	3.742738589	3.166244285

- According to this data set, when we select around 227,229,233 as bucket count we can see the average nodes in a bucket is less than 4. Now this satisfy the acceptability of only 3 unsucessful searches in average. Also standard deviation is around 2, this is also good value.
- But a downside is there are unused buckets since minimum number of nodes are 0. We can avoid it if we select buckets as 89 which has no unused buckets with a minimum standard deviation and average, then average nodes in a bucket will increase and misses in average will increase.
- Therefore we can fine tune the hashtable according to above methods according to our need.

Lets consider hash multiplication method,

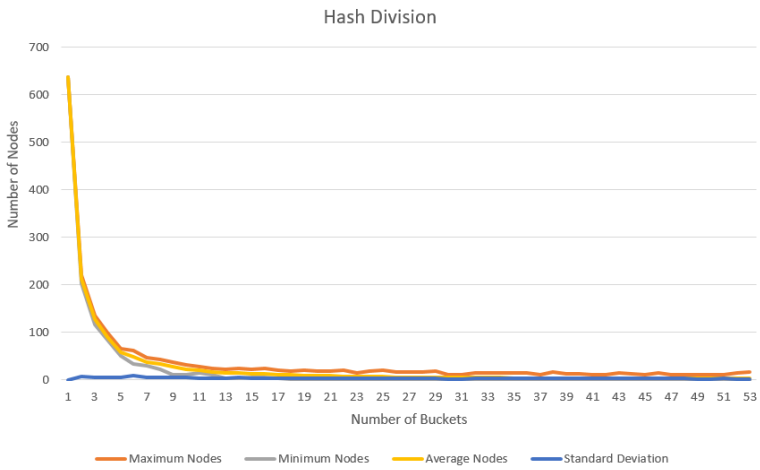


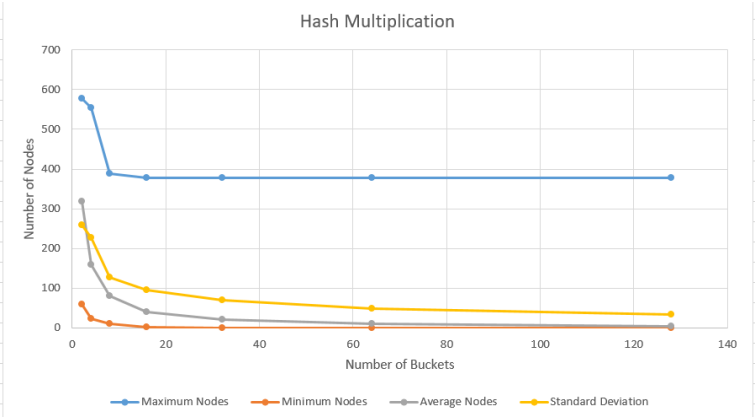
We can see in the plots there are no local spikes like in unimproved hash division method. Number of buckets in this method are powers of 2.

Usually in computers, it is easier to work with powers of 2. Therefore it is easier to make a hashtable with power of 2 number of buckets. Then the most suitable hashing method becomes the multiplication method. Division method is inefficient in powers of 2 table sizes. Also multiplication method is much faster when using with powers of 2.

Distribution for sample 2

Now lets check the distributions observed on Sample-text2.





Comparing distributions with sample 1

- Sample-text2 is a larger file than Sample-text1. Since distributions of Sample-text1 have higher values than this distributions, but they shows the same behaviour. Again we can observe hash multiplication is efficient at powers of 2 number of buckets while hash division is efficient at prime number of nodes.
- For both samples, division method seems to be have less values for standard deviation.
- Division method also have less values for maximum and minimum number of nodes in a bucket when considering bucket count.

For conclusion on best hash function for this particular purpose,

When comparing the 2 methods for this 2 sample files,

- Division method shows less standard deviation. Also it's maximum and minimum number number of nodes decrease rapidly than multiplication method when increasing the number of buckets.
- Also division method requires prime number of buckets while multipication method required powers of 2. Therefore division method can be implemented with many number of ways than multiplication.
- Therefore unless we have buckets which is a power of 2, division hashing method is suitable for this particular scenario.
- None of these 2 functions are idea hash functions because search time is not M/N where M is the number of keys and N is the number of buckets.