

small	words	Bins	occupancy	Used bin Fraction	Max entries in a bin
naive_hash	100	256	0.39	0.30	3
bernstein_hash	100	256	0.39	0.32	3
FNV_hash	100	256	0.39	0.33	4

medium	words	Bins	occupancy	Used bin Fraction	Max entries in a bin
naive_hash	2000	4096	0.48	0.18	12
bernstein_hash	2000	4096	0.48	0.38	4
FNV_hash	2000	4096	0.48	0.38	4

large	words	Bins	occupancy	Used bin Fraction	Max entries in a bin
naive_hash	83026	131072	0.633	0.012	283
bernstein_hash	83026	131072	0.633	0.469	7
FNV_hash	83026	131072	0.633	0.470	6

Comment:

As we can see from the result, the naïve hashing has the worse perform comparing two other methods. The fraction of used bin to the bin number is relatively low and the entries are gathered into several bin.

In my opinion, the reason why this happens is the difference between the calculation of the hashcode. For the Bernstein\_hashing, it multiplies the 'h' variable by '33' in order to differentiate the position of characters in a string. For the FNV hashing, it differentiates it with a larger long integer.

With the naïve hash, for example, the word 'abc' and 'cba' has the same hash code because the "weight" of each position is same. But with other two method, they differentiate the difference between positions, so they can levitate the extent of aggregation of entries.