

1. Any two words that contain the same letters in different orders would hash to the same values using the first hash function. An example would be "eat" and "ate".
2. String hash two is better than the first one because it causes more words to have distinct hash values, resulting in shorter linked lists and quicker searches for values.
3. No. Even if they hash to the same location, they are still two different words, and we check for words being different.
4. No. Same as above. The hashing value doesn't affect how many values there are or how big the table is.
5. Yes. With HashFunction1, you could end up with a mostly empty array with a bunch of values linked out into long lists, and have a mostly filled array of short lists with HashFunction2. In that case, the number of empty buckets would be very different between the two runs. A good example of this would be in storing a database of anagrams.
6. No. It just decreases because we are decreasing the size.
- 7.

Table size	runtime
1000	0.02
100	0.02
10	0.1
8	0.11
6	0.15
4	0.21
2	0.44

Table size v. Runtime without resizing

