

1. How good is the provided hash function -- are we really getting constant time operations with our hashmap?

The hash function is simple but will cause collisions. We could get the index by using the hash function. Therefore, the operation time for hashmap is not constant. If we have a large amount of data stored in this hashmap and when we search a certain item, the operation time increases as the size of data increases.

2. What is one other way you could implement the hash function? Anything creative is acceptable in this answer.

Maybe we could convert the input to hex and use loop to convert again to reduce to the chance of collisions.

3. If I have to resize the hashmap to add more buckets, what is the Big-Oh complexity?

The time complexity will reduce but the worst search operation time is still $O(n)$. If we still have the same amount of data and because there are more buckets in the array, the chance of collision will be reduced. The time operation complexity will decrease.

4. What is "open addressing" in regards to hash maps and hash tables?

Open addressing is a method to solve collision. By using open addressing, the $\text{index}(\text{address})$ of item could be different from the hash value. For example, when we insert item in bucket array and use hash function to get the index number. The index number could be same for two different item, we store item in the next available space in array. This method is called linear probing, and it is one of the open addressing method.