5.10

A separate chaining hash table is where you use a simple function to generate the key for each element you want to add to the table and then have each place be a linked list adding each value that collides at the spot. This method is good because it is easy to implement, and the hash table never fills up meaning that it’s great for when you don’t know how big the table needs to be or how many times there will be collisions.

Both linear probing and quadratic probing hash tables only allow one element per space in the table. Linear probing just looks for the closest available space when there is a collision by incrementing by one and wrapping back around until it finds the space. When there is a collision with a quadratic probing hash table the function uses a counter (usually i) that is squared and added to the element value to be used in the function again to find a new key. This continues until an available spot is found. Linear probing is better in terms of cache performance but as more elements are added consecutively it can lead to clustering making it take longer to find an available spot. Quadratic probing lessens the odds of clustering but has a worse cache performance than linear probing.

Double hashing uses a second hash function when there is a collision to find a new key in the table. This method has no clustering but has poor performance because it is computing two different functions. Cuckoo hashing also uses two hash functions but uses them to come up with two different keys for each element to decide the location it will go in two different tables. Also if there is a collision in a cuckoo table the current element will go into the spot and push the original element into the next table if there is also a collision there then it will completely push out what is there. Cuckoo hashing guarantees O(1) performance as long as the number of keys is not more than half the table size.