1a. The complexity of the hash function in respect to the string length in Big O notation is O(n). This is because the program has to do a calculation for each individual character of the string.

1b. The complexity of the search function involves the hash function. The hash function in respect to the length is O(n). But most likely the length of the string will be much smaller than the amount of buckets. For search the program has to hash first and then traverse the list of the hash. Hence in worst case scenario all the items will be in one list and the needed item will be at the end. So O(n) in respect to the number of items.

2.An ideal hash function must give an item the chance to be in any box. Meaning has to be mod sizeNumbr

An ideal hash function has to take into account the qualities of the item (in our case the string's length + the letters)

An ideal hash function cannot put a majority of the items in the same bucket.

An ideal hash function has a low complexity. Since this operation will occur every time.

3.I think my hash function is ideal because it gives every item a chance to be in any box rather than focusing them all in one area (mod size). It also takes into account the asci code of each letter in the word. It converts each character in the word to its asci code and sums them all together. The complexity isn't horrendously large either. And it spreads everything out.