A hash function is used by a hash table to determine where the value of the search key should be placed in the hash table. The hash function takes the search key and produces an integer that will be used as the index or location in the hash table where the value of the search key is stored. The hash function uses an algorithm to determine the integer, the most basic version of this, is that the hash function takes the last four digits of the search key and uses that for the index. The algorithms can become very complex though to try and avoid collisions, or instances where the hash functions happen to result in the same index.

A String hash function could be written where you convert the String into a hash key, one way to do this would be convert the String into its Unicode integers and then find the sum of those integers, the sum then being used as the hash key. If this produces too many collisions, then you would need to create an algorithm that increased the randomization or uniqueness of the hash key. This could be done by multiplying the characters together or taking the sum you previously found and factoring it by how many characters were in the String.

Using a hash function is generally considered to be a faster and more efficient process than searching for a key. The Big Oh of a search function is O(n) because a search function will need to use some type of a loop to iterate through all the values of the list to find the key. Whereas a hash function generally has a Big Oh of O(1) since it is just running one computation to find the value of the search key. The only time a hash function approaches O(n) is if there was a collision and then a search needs to occur, but if the algorithm is written to limit collisions and they are minimal to none then hash functions remain at a O(1) efficiency.

The Java Util HashMap uses a built-in method called hashCode() as its hash function for Strings. The method takes the Unicode of each character in the String, multiples it by factor based on its position in the String and then sums those products to create the hashCode.