

- **Problem 2**

- Hash table is suitable for single value lookup but not range scans because it does not store the keys in any order. If we want to find all values that are in a certain range, we need to scan the entire table.
- When the data size is extremely large, there's a high probability for hash collisions to occur and rehashing/table resizing is quite expensive and inefficient.  $O(1)$  is just average time for hash table operations but for trees, all operations are guaranteed to be done in  $O(\log N)$ .
- For BST, we can easily do in-order traversal to obtain all keys in a sorted order but it's impossible for hash table to do that.

- **Problem 3**

- A library of Python called "collections" provides us with a useful data structure: ordered dictionary. (Actually, a regular dictionary in Python 3.6+ is also "ordered" since it can remember the order of the keys, but the "collections" library allows us to use the function called "move\_to\_end" to move a key-value pair to the end of the dictionary, so it's more convenient. ) It's implemented using doubly linked list to maintain the order of elements.

Implementation available at:

<https://github.com/Wwwangi/STEP-2020/blob/master/HOMEWORK2/cache4.py>