Symbol Table documentation

Ghiurcuta Andrei-Bogdan 938pr

https://github.com/AndreiG23/FLCD

1a

Implementation

- I decided to use hash table as the data structure for the symbol table.The main advantage of an hash table over other data structure is speed.The access time of an element is on average O(1), therefore lookup can be performed very fast
- For each entry in ST a Deque(which is a double linked list) will be used. It ensures that will be no conflicts even when the elements hash to the same position, both of them will be in the same deque
- I use modular hashing : the hash function is a simple hash(k) = k % m for some m – size. The value k is an integer hash code generated from the key. And I use the sum of the ASCII code of each composing character to calculate the hash value

Supported operations:

- add(key):
  input: key = the token(string)
  output: getPosition(key)
- exists(key):
  input: key = the token(string)
  output: true/false
- getPosition(key):
  input: key = the token(string)
  output: (listPosition, listIndex) – listPosition is the position in the outer list and listIndex is the position in the current deque