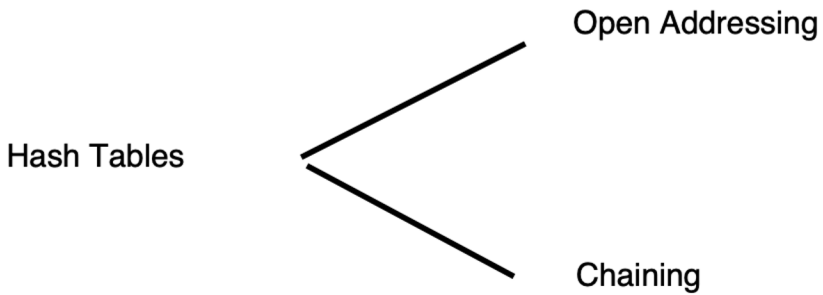
Hash Tables with Chaining

CSE 274

Prepared by Mahdi Ghamkhari

Hash Tables



Hash Table with open addressing

- We have already learned about open addressing option
- In open addressing, there is a single InternalArray to store keys
- We try to store a key at the corresponding hashIndex
- When a collision happens, we use the no-shifting mechanism to store the keys at an index after hashIndex

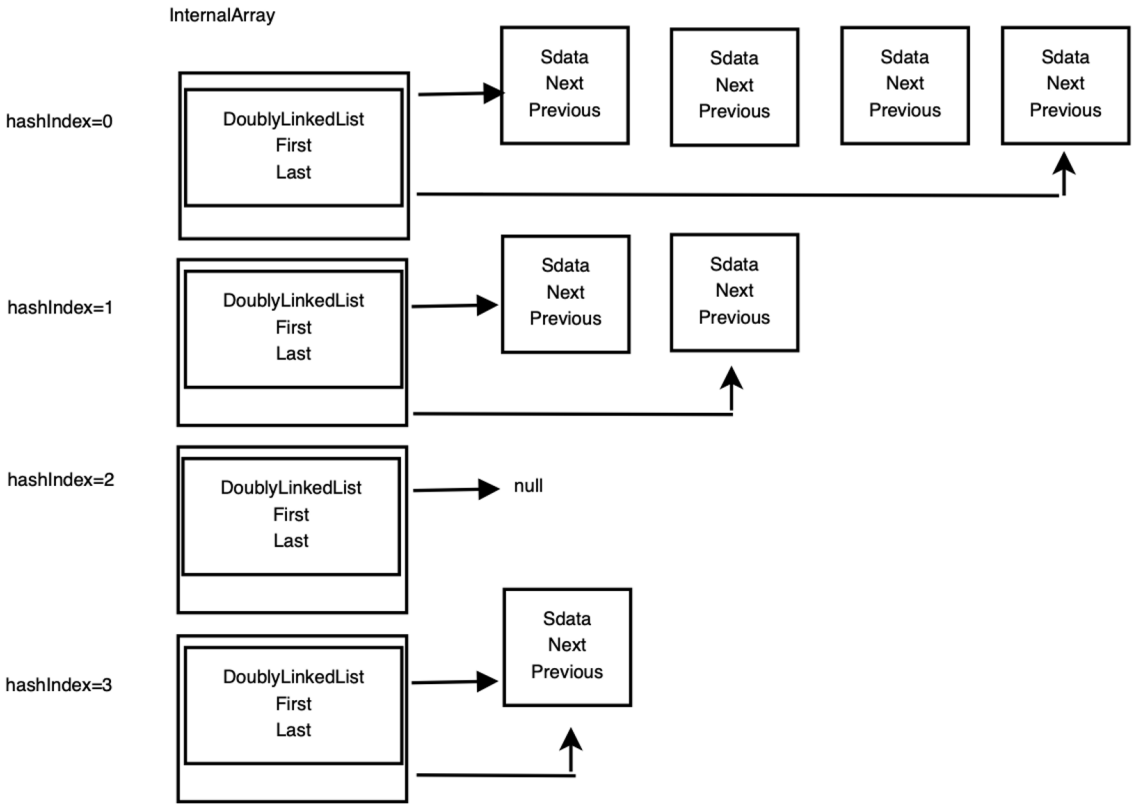
Hash Table with open addressing: performance

- The blocks of data merge but never split
- The length of blocks can become larger and larger
- The performance of insert, delete and find methods degrade as the number of data blocks and the length of data blocks increase

Hash Table with chaining option

- We still have blocks of data
- The blocks of data do not merge when using this option

Hash Table with chaining option



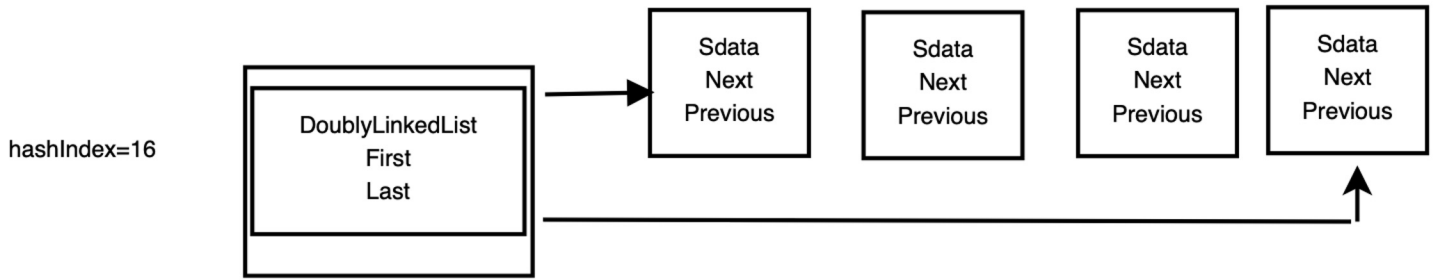
Hash Table with chaining option

- We have an InternalArray of size 52
- Each entry of the InternaArray stores a reference to an object of DoublyLinkedList class
- Each entry of the InternalArray stores a doubly Linked List

Inserting

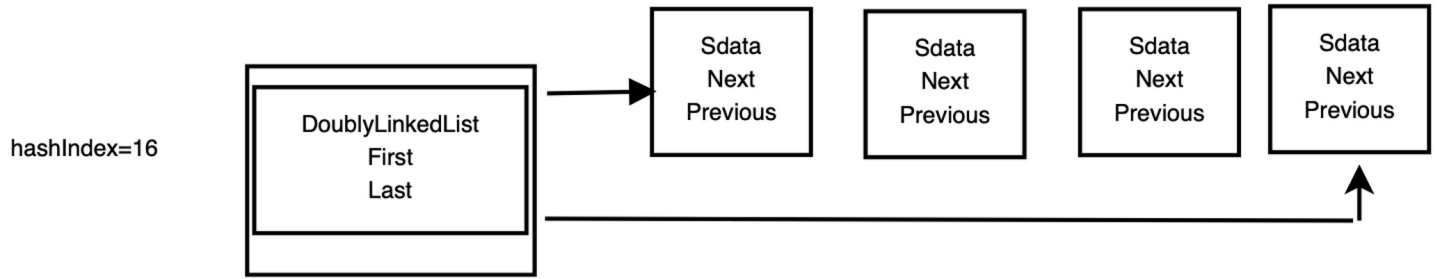
- Let say that we want to insert “FL” into the Hash Table.
- A hashIndex of 16 is obtained from hashFunc(“FL”)
- The InternalArray at hashIndex=16 is a reference to a Doubly Link List:

  InternalArray[hashIndex] ---- > a reference to a doubly linked list



Inserting

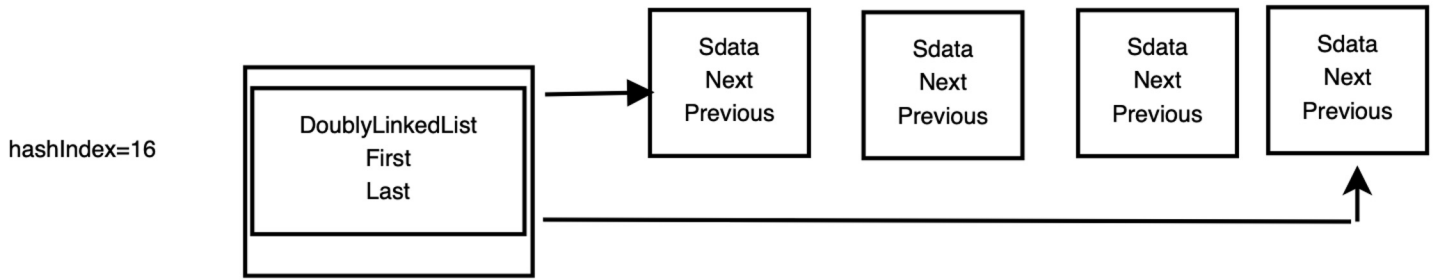
- Every object of the DoublyLinkedList class has insertNoDuplicate method
- We call InternalArray[hashIndex].insertNoDuplicate(“FL”);
- The insertNoDuplicate method checks if “FL” is already in Sdata field of one of the links. If “FL” was not found in the Links, a new Link containing “FL” is added to the linked list.



Deleting

- Let say that we want to delete “FL” from the Hash Table.
- A hashIndex of 16 is obtained from hashFunc(“FL”)
- The InternalArray at hashIndex=16 is a reference to a Doubly Link List:

  InternalArray[hashIndex] ---- > a reference to a doubly linked list



Deleting

- Every object of the DoublyLinkedList class has a delete method
- We call InternalArray[hashIndex].delete(“FL”);
- The delete method checks if “FL” is in Sdata field of one of the links. The delete method deletes such a link from the link list.