Remove Duplicates from an Unsorted Linked List

September 7, 2017

Question

Write a method void deleteDups(LinkedListNode n) which takes a pointer to a linked list (which may be unsorted) as an argument and returns void, but removes all duplicate nodes from the given linked list.

Explanation and Algorithm

Your algorithm should iterate through the linked list and store each node's data in a way that is easy to look up. A useful data structure for this is a hash table. Your algorithm should first check if the current node data exists in the hash table already. If it does, then you have found a duplicate node and must delete it. If it does not, then store the data and move on to the next node.

Hints

1. Would a buffer be useful? In what way could it be useful to be able to store data in your algorithm?
2. Are there any data structures you could use as a buffer? You should consider how easy/difficult it would be to look up information in your chosen data structure when you need to.
3. Try inputting your node data into a Hashtable.

Code

```
public static void deleteDups(LinkedListNode n) {
   Hashtable table = new Hashtable();
   LinkedListNode previous = null;
   while (n != null) {
      if (table.containsKey(n.data))
```

```
        previous.next = n.next;
      else {
        Table.put(n.data, true);
        previous = n;
      }
   n = n.next;
   }
}
```

Big O analysis

O(n): n is the number of nodes in the linked list; each node is visited exactly once.

Sources

Question, answer and other material taken from Cracking the Coding Interview 6th edition by Gayle Laakmann McDowell.