COMP2011

Lab and Tutorial Ten (Total Marks: 100)

Deadline 23:59, 27 Nov., 2014

1. Suppose the hash table is empty initially. Given the following integer sequence,

   8, 16, 24, 32,

   in each of the following questions, with different hash methods and array sizes, show *the final hash table* after the given integers have been inserted in order, and *the number of array cells* that have been gone through when searching each integer (see the example shown in the illustration slides). If the algorithm cannot terminate, show when this happens.

   (1) Hashing method: linear probing; Array size: 8

   (2) Hashing method: linear probing; Array size: 7

   (3) Hashing method: quadratic probing; Array size: 8

   (4) Hashing method: quadratic probing, Array size: 7

2. Suppose the hash table is empty initially, given the following integer sequence,

   15, 30, 45, 60, 75, 1

   in each of the following questions, with different hash methods and array sizes, show *the final hash table* after the given integers have been inserted in order, and *the number of array cells/links* that have been gone through when searching each integer. If the algorithm cannot stop, show when this happens.

   (1)Hashing method: double hashing; Array size: 15 (Second hash function: step = 5 – key%5)

   (2)Hashing method: double hashing; Array size: 7 (Second hash function: step = 5 – key%5)

   (3) Hashing method: separate chaining: Array size: 15

3. Download the lab10.java program that implements a double-ended linked list. Assume that iData of each link is equal to or greater than 0. Add the code in the method *removeDupLink( )* that can remove all duplicated links (a duplicated link is one that contains the same iData as one link in the linked list) and **runs O(n) where n is the number of links.** As discussed in the lecture, this can be achieved based on hash tables. For example, you can use class HashTable in the hash.java program attached.

What to submit:

(1) The report that contains (a) the results of Questions 1 & 2; (b) the output of your program from Question 3.

(2) The java source code of your program from Question 3.