CS 321 Data Structures (Fall 2020)

Homework #2 (86 points), Due date 11/16/2020 (Monday)

- **Q1(24 points): Basic Data Structures: Stacks, Queues, Linked Lists and Trees**

  (a)(8 points) Please write a pseudocode for `List-move(x, y)`. This procedure is to move an existing node x to the front of another existing node y in a doubly non-circular list without a dummy head.

  (b)(8 points) Given a binary tree T with a root node r, please write a recursive method `InternalNodes(r)` that returns the number of internal nodes with at least one child in the tree.

```
InternalNodes(r) // r is the root
{
  if r == null then return ________________________________________________;

  if (x.left == null && x.right == null)

     return _____________________________________________________;

  return ____________________________________________________;
}
```

(c)(8 points) How to use two stacks to implement a queue so that `enqueue` runs in $O(n)$ and `dequeue` runs in $O(1)$? Suppose the queues have no size limit. Please describe your algorithm without pseudocode.

- **Q2(20 points): Hashing**

  Suppose we would like to insert a sequence of numbers into a hash table with table size 8 using the three open addressing methods, with the primary hash function $h_1(k) = k \bmod 8$, the secondary hash function $h_2(k) = 1 + (k \bmod 7)$, and the constants $c_1 = c_2 = 1/2$ (in quadratic probing).

  (a)(10 points) If the sequence of numbers is $< 15, 39, 31, 63, 40 >$, please successively insert these numbers into the following tables.

index	linear	quadratic	double
0			
1			
2			
3			
4			
5			
6			
7			

  (b)(3 points) For the hashing functions and table size we used in part (a), does the linear probing fully utilize the table? How about the quadratic probing and double hashing ?

  (c)(7 points) A hash table with size 10 stores 6 elements. These 6 elements are stored in T[0], T[1], T[2], T[5], T[7], T[9]. Suppose that all the other entries contain no "deleted" flag. An entry has a "deleted" flag means that this entry stored an element before, but the element has already been deleted. If we would like to search an element with a key k and assume the linear probing technique is used, what is the expected number of probes for an unsuccessful search?

- **Q3(12 points): Expressions and Expression Trees**

  For a given in-fix expression as below:

  $$8 \div 2 \times 1 - 12 \div 6 \times 3 \div 2 + 4 * 7$$

  (a)(6 points) What is the corresponding expression tree?

  (b)(6 points) What are the corresponding pre-fix and post-fix expressions?

- **Q4(10 points): Binary Search Trees**

  For a given input array $A : < 7, 10, 5, 8, 3, 6, 2, 4, 1, 9 >$,

  (a)(6 points) What is the resulting binary search tree after inserting the numbers in the list to an initially empty tree?

  (b)(4 points) From the tree you have built in part (a), what is the resulting tree after deleting the value 7 ?

- **Q5(20 points): B Trees**

  (a)(10 points) For a sequence of keys {`j, a, i, b, h, c, g, d, f, e`}, suppose we would like to construct a `B-Tree`, with degree 2, by successively inserting those keys, one at a time, into an initially empty tree. Please draw the sequence of `B-Trees` after inserting each of the 10 keys.
  Note: please draw only one tree after each insertion.

(b)(5 points) Let t be the (minimal) degree of a BTree. Suppose the size of each object, including the key, stored in the tree is 50 bytes. Also, suppose the size of a BTreeNode pointer is 4 bytes. In addition, 30 bytes of meta-data is required for each BTree node to keep track of some useful information. Suppose each BTreeNode has only the meta-data, a parent pointer, a list of objects, and a list of child pointers. What is the optimal (minimal) degree for this BTree if a disk block is 4096 bytes?

(c)(5 points) For a BTree with height 4 (or 5 levels), what is the maximal number of objects can be stored if the (minimal) degree $t = 70$?