CS 5002 Practice Final Questions

Northeastern University— Seattle

Spring 2018

Answer the questions in the spaces provided on the question sheets. If you run out of room for an answer, continue on the back of the page.

Name: _____

1. What is one benefit of a linked list over an array for storing data?

Solution:

2. Consider the following problems. Give the complexity of an algorithm in big O notation. When necessary, briefly describe the algorithm. Give the answer using one of the following. O(1), O(log n), O(n), O(n logn), O(n^2), O(n!)
   (a) Finding an element in a sorted array using binary search _____
   (b) Reversing an array of integers _____
   (c) Inserting an element into the beginning of an array _____
   (d) Finding the longest increasing subsequence in an array_____

3. (a) Assume that L is a linked list of at least 2 nodes. We need to insert the node N (assume node reference is N) between head and the next node. Write the code fragments that can be used to do this.

Solution:

(b) Suppose we have a linked list of two nodes. Write a one line of code that will convert the linked list into a circular linked list.

Solution:

(c) What is the memory overhead of maintaining data as a linked list as opposed to an array?

Solution:

(d) Assume that we have an array of n integers and a linked list of n nodes each with an integer as data and a reference to the next node. Assuming that C pointers are 8 bytes, how much memory is needed to store the array of integers, and how much memory is needed to store the linked list of nodes?

Solution:

(a) Consider the following code

```
1 for (int i=0; i < n; i += 2)
2   for (int j = 0; j < n/2; j++)
3     Statement1;
```

How many times statement1 will be executed? Give the answer in Big O. _____

(b) List two advantages of using Linked Lists over Arrays

Solution:

(c) In each of the following cases, state whether Array or Linked list is the correct data structure to use.

The size of the data file is known in advance and many elements needs to be randomly accessed - _____

A List of bank records must be maintained and accessed in many different orders (i.e. sorted by name, age, account balance etc.) - _____

(d) Write a function insertNextTo(Node N, Node M) that inserts Node N next to Node M. You may assume the following node struct.

```
1 struct Node {
2   int data;
3   Node *next;
4 }
```

For example if the current list is

2 4 3 5 null

after calling `insertNextTo(7, 3)`

we have the list: 2 4 3 7 5 null

```
1 void insertNextTo(Node *N, Node *M) {
2
3
4
5
6
7
8 }
```

(e) Write a function `getNode(int n)` that returns the n-th node in the list. You must consider special cases such as n is larger than the size of the list, empty list etc. In all cases where a node cannot be found, return null.

```
1 public Node getNode(int n) {
2
3
4
5
6
7 }
```

(f) How many references you must change to delete a node from the middle of a singly linked list?

A. 1

B. 2

C. 3

D. 0

(g) Given an array and a singly linked list. Which of these data structures uses more memory space to store the same number of elements? Explain your answer.

Solution:

(h) Given a singly-linked list of unknown size. Describe in a few sentences (in English, please) how to find the middle element of the list without counting all nodes.

Solution:

(i) What changes do you need to make to a linked list in order to have a constant time access to the last node?

Solution:

(j) Given a sorted singly-linked list, where the head contains the smallest element:

Implement a function `insertInOrder(List *L, int newVal)` that creates a new node and inserts it in-order into the list. Assume a linked list:

```
1  struct list{
2    Node *head;
3  };
4  typedef struct list List;
5
6  struct node{
7    int value;
8    Node *next;
9    Node *prev;
10 }
11 typedef struct node Node;
```

```
1  void insertInOrder(List *l, int newVal){
2
3
4
5
6
7
8  }
```

(a) Given a collection of algorithms that runs on O(1), O(n log n), O(n), O(n2), O(log n), O(n!), order the algorithms from fastest to slowest

A. O(1), O(n log n), O(n), O(n2), O(log n), O(n!)
B. O(1), O(log n), O(n), O(n log n), O(n2), O(n!)
C. O(1), O(log n), O(n log n), O(n), O(n2), O(n!)
D. None of the above

(b) What is the best data structure to solve the following problem? A list needs to be built dynamically. Data must be easy to find, preferably in O(1). The user does not care about any order statistics such as finding max or min or median.

A. Use an Array
B. Use a Singly LL
C. Use a Stack
D. Use a Queue
E. None of the above

(c) Given a Queue Q, write a function that will find the max value in the queue. You may only use queue operations such as eneque, dequeue, head, tail. No other data structure can be used other than queues. The queue must remain intact after finding the max.

```
1  public Comparable findMax(Queue Q) {
2
3
4
5
6
7
8  }
```

(d) Write a function findLast that will find the last node of a LL. Return a pointer to the last node.

```
1 Node * findLast(LinkedList *myLL){
2
3
4
5
6
7
8 }
```

4. (a) Write a function `createArrayOfLL` that takes an array of words and creates an array of singly linked list nodes as follows:

   1. Create an array of 26 linked lists
   2. Insert each word into the array based on its first letter, eg: any word that begins with 'a' goes to A[0] list etc.
   3. Return the reference to the array of linked lists

   Consider ALL cases. Write reasonable comments to describe your algorithm.

```
1 Node * createArrayOfLL(char *[] words, int numWords) {
2
3
4
5
6
7 }
```

5. For each of the following scenarios choose the "best" data structure from the following list or a combination of data structures: an unsorted array, linked list, DLL, stack, queue. In each case, justify your answer briefly.

   (a) Some large dictionaries and encyclopedias have thumb tabs for each letter, cut-outs in the edge of the volume so the reader can turn directly to the first page of listings for that letter. Is this access to the beginning of each letter's listings more like a stack, queue, array, tree, or linked list?

   _____

   (b) When cookbooks describe complicated recipes, they break them into sub-recipes, much like procedures in a programming language. Thus, the recipe for a cake might say, "Use the chocolate icing recipe on page 23," and that chocolate icing recipe might say in turn, "See page 195 for instructions on melting chocolate." Which data structure would you use to represent the sequence of recipes and sub-recipes being carried out at a given moment, to make it most convenient to return to the "calling" recipe when each sub-recipe is completed: a stack, queue, array, tree, or linked list?

   _____

   (c) Is a book's table of contents, with chapters, sections, and sub-sections, more like a stack, queue, array, tree, or linked list?

   _____

   (d) Most newspapers run a new crossword puzzle every day. Below the puzzle it generally says, "Solution in tomorrow's newspaper." Is this sequence of puzzles and solutions more like a stack, queue, array, tree, or linked list?

   _____

(e) Some people are very rigid about reading newspapers in chronological order; they won't read one day's newspaper unless they've read all the previous days' papers, in order. Even if days or weeks go by when they don't have time to read the paper, they'll save the papers, in order, and read them in order when time permits. Is this arrangement more like a stack, queue, array, tree, or linked list?

_____

(f) Suppose that a grocery store decided that customers who come first will be served first _____

(g) A list must be maintained so that any element can be accessed randomly _____

(h) A program needs to remember operations it performed in opposite order _____

(i) The size of a file is unknown. The entries need to be entered as they come in. Entries must be deleted when they are no longer needed. It is important that structure has flexible memory management. _____

(j) A list must be maintained so that elements can be added to the beginning or end in O(1) _____

6. Draw the contents of the hash table below. Show work. The size of the table is 3. The hash function is $H(k) = k \mod 3$.

   Insert:

   9, 15, 2, 0, -1, 5, 29

Table 1: Hashtable with chaining

0	
1	
2	

7. What is $41^{65} \mod 7$)? _____

8. Show that any for any integer a, $a^3 = a \mod 6$ _____

9. What is $2^{12} \mod 13$ and what is $2^{11} \mod 12$) and what is $2^{14} \mod 15$)(Hint: Fermat's Little Theorem) _____

10. Let R be the relation on $A = \{1, 2, 3, 4\}$ defined by "x is less than y". Write R as a set of ordered pairs.

11. Determine whether the relation R on the set of positive integers is reflexive, transitive, symmetric, asymmetric, antisymmetric, where $(x, y) \in R$ if and only if:

    (a) $x > y$ _____

    (b) $x + y = 10$ _____

    (c) $x + 4y = 10$ _____

12. A common problem for compilers is to determine whether parens balance. For example, the string "((())())()" contains properly nested parens, while ")()(" and "())" do not. Give an algorithm (pseudocode) that returns true if a string contains properly nested and balanced parens, and false if not.

Solution:

13. Write a program to reverse the direction of a given singly-linked list. That is, after the reversal, all the pointers should now point backwards. The algorithm should take linear time.

Solution:

14. What method would you use to look up a word in a dictionary? (what data structure, and briefly, what steps?)

Solution:

15. Write a function to find the middle node of a singly-linked list.

Solution:

16. Write a function to compare whether two binary trees are identical. Identical trees have the same value at each position and the same structure.

Solution:

17. What is the best data structure for maintaining URLs that have been visited by a Web crawler?

Solution: