

# COMP 272

## Data Structures and Algorithms

### Sample Final Examination (Rev. 7)

#### Instructions to the student:

1. You will be allowed **three (3) hours** to complete the examination.
2. This is an **online** and **closed-book examination**. Only materials distributed with this examination may be used by the student to answer examination questions. You are **NOT** allowed to use textbooks, workbooks, notes, tapes, cell phones, hand-held computers or laptop computers, other electronic/digital devices, or to consult with other people while writing this examination.
3. This examination has four parts and the structure of the examination is detailed below.
  - Part 1 (40%): Multiple-choice questions (20 questions, 2 marks each)
  - Part 2 (35%): Fill in the blanks (10 blanks, 3.5 marks each)
  - Part 3 (15%): Short answer questions (3 questions, 5 marks each)
  - Part 4 (10%): Programming question (1 question, 10 marks)
4. To answer a multiple-choice question, choose the letter corresponding to the best alternative provided. There is no penalty for guessing so you should attempt every question even though you may not be certain of the answer. If you choose more than one alternative, the answer will be marked wrong – except those problems where it is explicitly indicated otherwise.
5. Note that one of the requirements for obtaining three credits in Computer Science 272 is that you **MUST** score at least 50% (50 marks) in this examination.
6. Legibility is important; an examination that cannot be read cannot be graded.
7. Complete the information requested on the **Final Grade Report** included in your examination package.

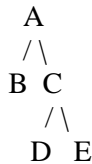
# COMP 272

## Data Structures and Algorithms

### Sample Final Examination (Rev. 7)

#### Part 1: Multiple-choice questions (20 questions, 2 marks each)

1. Which of the following traversals yields BADEC?



- a. only in-order
  - b. only level order
  - c. only post-order
  - d. only pre-order
  - e. pre-order and level order
  - f. in-order and level order
  - g. none of the above
2. Which of the following is a post-order traversal of the BST?
- a. ACEDB
  - b. ABDCE
  - c. BDECA
  - d. EDCBA
  - e. BADCE
  - f. BADEC
  - g. one of the above
3. Given `current` (a reference to a list node), which statement may insert an item `x` correctly after the node referenced by `current`?
- a. `current = new ListNode( x, current );`
  - b. `current.next = new ListNode( x, current.next );`
  - c. `current.next() = new ListNode( x, current );`
  - d. `current.next() = new ListNode( x, current.next );`
  - e. `current = new ListNode( x, current.next );`
  - f. `current.next = new ListNode( x, current );`
  - g. none of the above
4. Two sorting algorithms whose worst-case running times are in  $O(n \log n)$  are
- a. merge-sort, Shellsort
  - b. heap-sort, quicksort
  - c. heap-sort, Shellsort

- d. insertion sort, merge-sort
- e. heap-sort, merge-sort
- f. merge-sort, quicksort
- g. none of the above

5. The subarray of length 7 shown below is about to be partitioned by quicksort.

6 8 4 5 3 1 7

The pivot is selected by the “safe choice”; that is, it will be the middle element, 5. The pivot is then swapped with the last element, 7, and partitioning starts. At the end of partitioning, the pivot is swapped into place. At this point the state of the subarray is

- a. 1 3 4 5 6 8 7
  - b. 1 3 4 5 8 6 7
  - c. 1 4 3 5 6 8 7
  - d. 1 4 3 5 8 6 7
  - e. 1 3 4 5 6 7 8
  - f. 1 3 4 5 7 6 8
  - g. none of the above
6. Recursion is sometimes preferred to iteration because it
- a. is generally faster than iteration.
  - b. requires less memory space.
  - c. is based on proof by contradiction.
  - d. may increase the complexity of the solution.
  - e. is rooted in transfinite induction.
  - f. is based on the pigeonhole principle.
  - g. none of the above
7. Linked lists are often implemented with a dummy header node to
- a. reduce the execution time of the list operations.
  - b. fulfill the specifications of the list ADT.
  - c. identify the beginning of the list.
  - d. be able to access the first element in  $O(1)$  time.
  - e. reduce code complexity.
  - f. save memory space.
  - g. none of the above
8. When an unordered list of  $n$  keys is searched sequentially for given key values, the average number of key comparisons made by an unsuccessful search is
- a.  $n$
  - b.  $\log n$
  - c.  $n/2$
  - d.  $n \log n$
  - e.  $n + n^2$
  - f.  $n^2 \log n$
  - g.  $2n$

9. Which data structure would you use in writing a program to check for balanced parentheses?

- a. binary search tree
- b. hash table
- c. priority queue
- d. queue
- e. stack
- f. list
- g. none of the above

10. Which of the following functions is not  $O(\log(N))$ ?

- a.  $\log(\log(N))$
- b.  $1000 + \log(N)$
- c.  $1000 \log(N)$
- d.  $\log(1000N)$
- e.  $\log(N^2)$
- f.  $1000 \log(1000N^{1000})$
- g. all of the above are  $O(\log(N))$

11. And ten more questions in Part 1 on the final examination.

...

## Part 2: Fill the blank questions (10 blanks, 3.5 marks each)

21. Given data structure D that represents connected nodes N1, N2, and N3 shown in the (real) examination question. Write pseudo-code in (blank 1) to implement the function of changing the connection between given nodes N1, N2, and N3. The running time of the function is (blank 2)

22. Given a stack of integers, S, a queue of integers, Q, and the following a series of stack and queue operations.

```
S.push(11);
S.push(12);
S.push(36);
S.add();
Q.enqueue(S.pop());
Q.enqueue(S.pop());
Q.enqueue(25);
Q.mult();
```

Method `add()` pops the stack twice and pushes the sum of the two popped items back onto the stack. Method `mult()` dequeues two elements from the queue and enqueues their product in the same queue. If initially S and Q are empty, then after the above sequence of messages the content of queue Q (in front-to-rear order of the queue elements) is (blank 3)

23. And eight more questions in Part 2.

### **Part 3: Short-answer questions (3 questions, 5 marks each)**

31. Explain the concept of *binary search tree*.
32. And two other concept-related questions

### **Part 4: Programming question (10 marks)**

34. Describe, in pseudo-code, an algorithm for finding a node with a specific data value in a given data structure type by means of a given traversal method starting from a specific node.



Please note, your real examination question will provide specific data values, data structure (type), required traversal method, and a specific node to be searched based on the algorithm you will implement in pseudo-code.

END