**CSE214**

# EXAMS

EXAM PRACTICE QUESTIONS – MIDTERM 1

DISCLAIMER
THESE ARE MERELY PRACTICE PROBLEMS FOR YOU TO WORK ON. THE ACTUAL
MIDTERM MAY NOT ASK THESE QUESTIONS IN THE SAME WAY AND MAY INCLUDE
OTHER TOPICS THAT WE HAVE DISCUSSED IN CLASS, COMPLETED FOR RECITATION
AND HOMEWORK, OR DISCUSSED IN YOUR TEXTBOOK.

1. Consider the code fragment below.

```
for (x = 1; x <= n; x++)
{
    a = a + x;
    for (y = 1; y <= x; y++)
        b = b + y;
}
```

Counting only assignment statements as operations,

(a)  exactly how many operations does this code execute in terms of n? Express this
answer in closed form.

(b)  what is the order of complexity of this code in terms of n? Express this answer as a
Big O function in its simplest form.

2. A program runs on a supercomputer to analyze weather data from N weather stations. The
original execution time of the program is 10 minutes. If the amount of data is increased to 16N,
approximately how long in minutes will it take to run this program if the program has the following
number of operations?

(a) $O(N)$

(b) $O(N^2)$

(c) $O(N \log_2 N)$

3. Complete the following method for IntList assuming IntList is a singly–linked list with a
reference to the head of the list only.

```
public void moveFirstToLast() {
```

```
// moves the first element of the list to the end of the list
// by changing data values only

    IntNode nodePtr = _____;

    if (_____) {

        // List has 0 or 1 element. No work to do. Just return.

        return;

    } else {

        // save data from first node

        int temp = _____;

        // copy data values into previous nodes, one by one

        while (_____) {

            nodePtr.setData(_____);

            nodePtr = _____;

        }

        // store head data into last node

        nodePtr.setData(_____);

    }

}
```

4. Complete the following method for IntList assuming IntList is a <u>circularly</u>–linked list with a reference to the <u>head</u> of the list only.

```
public void moveMaxToHead() throws EmptyListException {

// Swaps the maximum element of the list with the head element.

    IntNode nodePtr = head;

    IntNode maxPtr = head;      // pointer to node with max value

    int temp;

    if (_____) {   // is list empty?

        _____;

    } else {   // search list, setting maxPtr to node with max value
               // until you examine the node before the head node

        while (_____) {

            nodePtr = _____;   // advance one node

            if (_____)

                maxPtr = nodePtr;
```

```
        }

        // swap first and max values

        temp = _____;

        head.setData(_____);

        maxPtr.setData(_____);

    }

}
```

5. A list can be implemented as an `IntList` that uses a linked sequence of `IntNodes` or using an array of type `int`. Assume that an `int` takes up 4 bytes in memory, a reference takes up 4 bytes in memory, and an array A is defined to hold up to 200 `ints`.

   (a)  If a list has 180 integers in it, which method (array or linked list) is more efficient in terms of memory usage?

   (b)  If a list has 20 integers in it, which method (array or linked list) is more efficient in terms of memory usage?

   (c)  The array and linked list implementations require the same amount of memory if the list has how many elements?  (Do not include `head` in this calculation.)

6.  What is the output of the following algorithm (a) if x = 2?  (b) if x = 3?

        **Let S be an empty intStack.**

        **y = 7**

        **while y > 0 do**

                **if (y mod x != 0)  S.push(y)**

                **else  output S.pop()**

                **y = y - 1**

        **while not S.isEmpty()**

                **output S.pop()**

7. Use the algorithm discussed in class to convert the following infix expression into postfix notation by showing the state of the operator stack at each point in the conversion.

$$A * B / (C + D)$$

8. Let `s` be a stack that holds data of type `Object`. Assuming `a` is of type `Location`, explain if each statement is correct or not, and why.

(a)   `S.push(a);`

(b)   `a = S.pop();`

9. Let `SortedIntList` be a class that uses `IntNode` nodes to create a <u>singly–linked</u> list such that the data is stored in <u>decreasing</u> order in the list. Assume that the class has two private instance variables `head` and `tail` of type `IntNode`. Write the following methods:

- **`public boolean remove(int item)`**
  Removes the item from the sorted list, if it exists, and returns true. Otherwise, returns false.
- **`public int maximum()`**
  Returns the maximum element in the sorted list, if the list is not empty. Otherwise, throws an exception.

10. Complete the following method in Java based on its specification:

```
public int evaluate(String postfix)

      Evaluates the given postfix string using a stack.

       Preconditions:


      - The operands in the postfix string are single-digit integers.

      - The postfix string contains only operands and operators +,-,*, and /.
         (No spaces.)

       Postconditions:

      - Returns the integer value of the given postfix string if the string is a
         valid postfix string.

       Notes:
       - Integer division is used.
       - You may assume the input string is a valid postfix expression.
```

*Special note:*
*There are 10 possible digits in the string, but you can convert these to integers in the*
*same way. Do not write a huge* `switch` *statement or* `if` *statement with 10 options!* <u>*THINK!*</u>

---

**Course Info** | **Schedule** | **Sections** | **Announcements** | **Homework** | **Exams** | **Help/FAQ** | **Grades** |
**HOME**