CSE214

EXAMS

EXAM PRACTICE QUESTIONS – PART 1

THIS REVIEW PAGE DOES NOT IMPLY THAT THE ACTUAL MIDTERM QUESTIONS WILL BE OF THE SAME FORMAT.

```
1.
(a) n + (1+2+...+n) = n + n(n+1)/2
(b) O(n^2)

2.
(a) 10 * 16N/N = 160 min
(b) 10 * (16N)^2/N^2 = 2560 min
(c) 10 * (16Nlog16N)/NlogN = 160*log16N/logN min
        (depends on what N initially was)

3.
(a) head
(b) head == null || head.getLink() == null
(c) head.getData()
(d) nodePtr.getLink() != null
(e) nodePtr.getLink().getData()
(f) nodePtr.getLink()
(g) temp

4.
(a) nodePtr == null (or maxPtr == null)
(b) throw new EmptyListException("error message")
(c) nodePtr.getLink() != head
(d) nodePtr.getLink()
(e) nodePtr.getData() > maxPtr.getData()
(f) head.getData()
(g) maxPtr.getData()
(h) temp

5. (not counting the "head" variable in this case)
(a) List: 180*8 = 1440 bytes    Array: 800 bytes        Array is better
(b) List: 20*8 = 160 bytes      Array: 800 bytes        List is better
(c) n*8 = 800, so n=100


6.
(a) 7 5 3 1
(b) 7 4 1 2 5
```

```
7.
OPERATOR STACK          POSTFIX EXPRESSION
$
$                       A
$ *                     A
$ *                     A B
$ /                     A B *
$ / (                   A B *
$ / (                   A B * C
$ / ( +                 A B * C
$ / ( +                 A B * C D
$ /                     A B * C D +
                        A B * C D + /

8.
(a) OK, since the type Location is widened to Object, its parent class,
    automatically in Java
(b) NO, the data type returned will be of type Object, which must be
    narrowed to Location before it is stored in "a".  This can be done
    using typecasting.

9.
(a)
public boolean remove(int item) {
        IntNode cursor;
        if (head == null) return (false);
        else if (head == tail) { // 1 node list
                if (head.getData() == item) {
                        head = null;
                        tail = null;
                        return (true);
                }
                else return (false);
        }
        else { // 2 or more nodes in list
                if (head.getData() == item) {
                        head = head.getLink();
                        return (true);
                }
                cursor = head;
                while (cursor.getLink() != null) {
                        if (cursor.getLink().getData() == item) {
                                cursor.setLink(cursor.getLink().getLink());
                                if (cursor.getLink() == null)
                                        tail = cursor;
                                return (true);
                        }
                        else if (cursor.getLink().getData() < item)
                                return (false);  // can't be there
                        else
                                cursor = cursor.getLink();
                }
                return (false);  // not found in entire list
        }
}

public int maximum() throws Exception {
        if (head == null) throw new Exception("list is empty");
        else return (head.getData());
}

10.

public int evaluate(String postfix) throws DivisionByZeroException {
        int length = postfix.length();
        char ch;
        int index, value;
        int operand1, operand2;
        IntStack S = new IntStack();

        for (index = 0; index < length; index++) {
                ch = postfix.charAt(index);
                if (Character.isDigit(ch)) {
                        value = (int) ch - (int) '0';
                        S.push(value);
                }
                else {
                        operand2 = S.pop();
                        operand1 = S.pop();
                        if (ch == '+')
                                S.push(operand1 + operand2);
                        else if (ch == '-')
                                S.push(operand1 - operand2);
                        else if (ch == '*')
                                S.push(operand1 * operand2);
                        else if (ch == '/') {
                                if (operand2 == 0) throw new
                                        DivisionByZeroException("err msg");
                                S.push(operand1 / operand2);
                        }
                }
        }
        return S.pop();
}
```