

Part A There are 8 **multiple-choice** questions in this part.

You **must** use the supplied multiple-choice answer sheet to record your answers for this part (Part A). Only answers on that sheet will be considered (answers on the exam booklet itself will be **discarded**). You must highlight your ID on that multiple-choice sheet as well. A correct choice for one question will get you **4** points. An incorrect answer, marking several answers or selecting nothing for a question will get you **0**. If you believe that more than one answer is correct, select the best answer.

1. What is the best asymptotic (“big-O”) characterization of the following function:

$$f(n) = (14\log n)^2 + \log(3^n)$$

- a) $O(3^n)$
 - b) $O(n^2)$
 - c) $O(n)$
 - d) $O(2^n)$
 - e) $O(\log n)$
2. Give the best asymptotic (“big-Oh”) characterization of the worst case *and* the best case time complexities of the algorithm *DoAgain*(*A*, *n*) .

Algorithm *DoAgain*(*A*,*n*)

Input: Array *A* storing integers and of size $n > 1$.

```

sum ← 0
for c ← 0 to  $n^2$  do
  if  $A[0] < 0$  then
    for k ← 0 to  $n - 1$  do
      sum ← sum +  $c \cdot A[k]$ 

```

- (a) Best case $O(n)$ and worst case $O(n^2)$
 - (b) Best case $O(n)$ and worst case $O(n^3)$
 - (c) Best case $O(n^2)$ and worst case $O(n^3)$
 - (d) Best case $O(n)$ and worst case $O(n^4)$
 - (e) Best case $O(n^3)$ and worst case $O(n^4)$
3. You have implemented the queue with a linked list, keeping track of a front node and a rear node with two reference variables. Which of these reference variables will change during an insertion into a NONEMPTY queue?
- a) Neither changes
 - b) Only front changes.
 - c) Only rear changes.
 - d) Both change.

4. Here is an incorrect kind of pseudo code a student provided for the algorithm which is supposed to determine whether a sequence of parentheses is balanced:

```

declare a character stack
while ( more input is available)
{
    read a character
    if ( the character is a '(' )
        push it on the stack
    else if ( the character is a ')' and the stack is not empty )
        pop a character off the stack
    else
        print "unbalanced" and exit
}
print "balanced"

```

Which of these unbalanced sequences does the above code think is balanced?

- a) (())
- b) ()) (()
- c) (())())
- d) (())) (

5. Consider the algorithm *Multiply*(*A*,*n*) below. *A* is an array of size *n* storing integer values. What is the best characterization of the best and worst case asymptotic time complexity of the following algorithm?

Algorithm *Multiply*(*A*,*n*)

```

    result = 0
    for i = 0 to  $\frac{n}{2}$  do
        j = 0
        while  $A[j] < 0$  and  $j < \frac{n}{2}$  do
            result = result +  $A[i] * A[j]$ 
            j = j + 1
    return result

```

- a) Best case $O(1)$, worst case $O(n)$.
- b) Best case $O(\log n)$, worst case $O(n)$.
- c) Best case $O(n)$, worst case $O(n)$.
- d) Best case $O(1)$, worst case $O(n^2)$.
- e) Best case $O(n)$, worst case $O(n^2)$.

6. Consider the following pairs of functions: $f(n), g(n)$. For which pair, the functions are such that $f(n)$ is $O(g(n))$ and $g(n)$ **is not** $O(f(n))$?

- (a) $f(n) = n^2, g(n) = 2n^2 + 7$
- (b) $f(n) = \log n, g(n) = \log(n^2)$
- (c) $f(n) = 17, g(n) = n + 1$
- (d) $f(n) = n, g(n) = 3000n + 1$
- (e) $f(n) = \log n, g(n) = 1/n$

7. What is the best asymptotic ("big-O") characterization of the following function:
 $f(n) = 6n \log \log(n^2) + 2n \log^2 n + n \log n$

- a) $O(n \log^2 n)$
- b) $O(n \log n)$
- c) $O(n^2 \log n)$
- d) $O(n^3 \log n)$
- e) $O(n \log(n^2))$

8. Suppose we have a circular array implementation of the queue class, with ten items in the queue stored at data [2] through data [11]. The current capacity is 42. Where does the insert method place the new entry in the array?

- a) data[1]
- b) data[2]
- c) data[11]
- d) data[12]

Part B**Important Note**

- ❖ Efficiency refers to time and memory.
- ❖ You will be graded as follows:
 - A mark (100%) if you provide a **correct and most efficient** algorithm.
 - A mark (%80) if you provide a **correct and second efficient** algorithm.
 - A mark (%50) if you provide a **correct** and missed to consider the efficiency of the algorithm.

B.1 [33 Points] A palindrome is a string that reads the same forward and backward, capitalization and space are ignored. For example *deed*, *go dog*, *level*.... are palindromes.

- a) **[10 Points]** Write an iterative algorithm in pseudo code that tests whether a string is a palindrome.
- b) **[10 Points]** Write a recursive algorithm in pseudocode that tests whether a string is a palindrome.

ID:

- c) **[8 Points]** Describe how you could use a *Stack* or *Queue* to check whether a string is a palindrome, and write the pseudo code for that.

- d) **[5 Points]** What are the worst-case runtimes of above algorithms (a), (b), (c) (use the big-Oh notation). Justify your answers.

Part C [24 Points]

For each of the 4 questions in this part, mark **T** if the given statement is **ALWAYS** true. Otherwise mark **F** and **justify** your answer. If you do not justify the FALSE case you will lose $\frac{4}{6}$ of the mark. There is **no penalty** for selecting a wrong answer. **Hint:** a correct counter example and/or correct specification will give you better marks. A correct answer will get you **6** points.

1. If $f(n) = 5n^2$ then $f(n) \in \Omega(2^n)$ ☐T ☐F
2. The worst-case asymptotic running time for the best algorithm for finding something in a sorted array? is $O(n)$ ☐T ☐F
3. The worst-case asymptotic running time of finding and removing all values greater than 12 from a stack implemented with a linked-list (leaving the rest of the stack in its original order) is $O(1)$ ☐T ☐F
4. Performing a remove operation at the tail in a list ADT implemented as a singly LinkedList that keeps track of the head, is very efficient: performed in a constant time $O(1)$. ☐T ☐F