
This exam contains 8 pages (including this cover page) and 9 questions. Total of points is 100.

Conditions: You are permitted writing utensils and the midterm. No other items are permitted, e.g. no notes, no text,

Note: The pages are double-sided. **PLEASE WRITE your name** on all pages.

I, _____, understand the above statements and agree to follow these terms, and upon my honor, I swear that the answers provided are of my design and of my effort alone. I have not received nor viewed answers from any source but myself.

< sign > _____

1 Complexity (10 minutes)

1. (18 points) Computational Complexity.

- A. Given the following pseudocode, determine the worst case Big-Theta time and space complexities, in terms of n . Identify each separately.

```
sum := 0
for i from 1 to n
  for j from 1 to n
    for k from 1 to n
      sum := sum + i * j
```

- B. Given the following pseudocode, determine the worst case Big-Theta time and space complexities, in terms of n . Identify each separately.

```
i := n
while(i > 1)
  i := i/2
  print(i)
```

- C. Given the following pseudocode, determine the worst case Big-Theta time and space complexities, in terms of n . Assume `listNode` is a pointer to a node in a linked list of length n . Assume the function is initially called using the head of the list. Identify time and space complexity separately.

```
function search( listNode, val )
  if listNode is NULL return NULL
  else if listNode.data == val, return listNode
  else search(listNode→ next, val )
```

2 Data Encoding (10 minutes)

2. (12 points) Data.

A. Briefly describe how data is encoded in a computer.

B. What is the binary representation of the int value 9 assuming an 8-bit representation of ints?

C. Assume $\&$, $|$, \ll , and \gg are *bitwise* symbols / operators to perform the following operations on binary encodings (respectively): AND, OR, left-shift, right-shift. Using pseudo-code or c-like code, implement an *efficient* **function** `doubleVal(int x)` that returns the int value $2 * x$. You cannot use the multiplication operator `*`.

3 Circular Dynamic Array (10 minutes)

3. (10 points) In general, an array is limited due to its contiguous allocation. A dynamic array provides an abstraction that conceals this constraint (but not does circumvent it.) Briefly describe how a dynamic array can “change” size dynamically. (Be precise and concise: 2-3 sentences should suffice)
4. (5 points) Consider a circular, dynamic array structure with member attributes `int numItems`, `int first`, `int last`, and `array` (which is THE array member). Consider the valid states of this structure. Using pseudocode or c-like code, implement a **function `CDArray::removeFirst()`** that effectively “removes” the first item in the circular dynamic array. Hint: Be sure a valid state of the structure is maintained.

4 Polynomial (10 minutes)

5. (15 points) In your project, describe how you implemented the polynomial structure. Include all structure attributes (member variables and types only – no need to include methods) and comprehensively (yet concisely) describe each member and what it represents. Feel free to use an illustration or UML diagram to help support your discussion.

5 Tree (20 minutes)

6. (15 points) Consider the following tree with nodes A,B,C,D,E,F, and G: where C has children D and E; G has children A, B, and C; and A has child F.

A. Draw an illustration of the tree where nodes are represented as circles and edges as lines.

B. Represent this tree using set and tuple notation.

C. What is the root of this tree?

D. What are the leaves of this tree?

E. What are the descendants of A?

7. (10 points) Consider a Tree structure with attribute: Node* root. Also consider a Node structure with attributes: Node* leftChild, Node* rightChild (only two children per node). Using pseudocode or c-like code, implement a **function heightOfTree(root)** that returns an int which is the height of the tree. If you prefer to provide pseudocode for a generic tree (rather than a binary tree with only two children – that is OK). *Hint: Consider the recursive definition of a tree. How is the height of a tree defined in terms of the height of its subtrees?*

6 Stack (15 minutes)

8. (10 points) Consider the stack data structure. Assume you are tasked with designing an *efficient* data structure called DoubleStack, which consists of two separate stacks. *Also assume that memory use is highly restricted – only n memory locations, in total, can be used for the two stacks.* For example, stack #1 might have $n - i$ items and if stack #2 has i items, then no more items could be added to either stack. Assume the operation to be performed on a DoubleStack ds are as follows: DoubleStack::push(int stackNum, int val) and DoubleStack::pop(int stackNum) such that ds.push(1,10) would push 10 onto stack #1, ds.push(2,15) would push 15 onto stack #2, ds.pop(1) would perform a pop on stack #1 and ds.pop(2) would perform a pop on stack #2. Since we are restricted to n total spaces for both stacks, if all n spaces are full, we will assume the push function would be unsuccessful and throw an exception/error. Assume you are tasked with designing DoubleStack, explain the design and describe implementation details of your structure DoubleStack. Include all attributes (member variables). Discuss the details of how the stacks will be stored in memory, how the restriction of n memory locations will be maintained, and what member variables are necessary to maintain this structure. Feel free to use an illustration and/or a UML diagram to help in your explanation. *A goal of this design is to efficiently make use of the n memory locations.* An efficient design would ensure that if any of the n spaces are free, push(1, val) or push(2, val) would be successful.

9. (5 points) Using pseudocode or c-like code, implement a method **DoubleStack::push(int stackNum, int val)** with the following behavior: `doubleStack.push(1, 10)` will push the value of 10 onto stack number 1 and `doubleStack.push(2, 15)` will push the value of 15 onto stack number 2. If an item cannot be added, given the memory constraint, the function will execute “throw error”.