CS251 - Data Structures and Algorithms
Fall 2024

PSO 5, Week 6

```
function ExchangeSort(A : array)
    let n be the size of A
    for i from 0 to n − 2 do
        for j from i + 1 to n − 1 do
            if A[i] > A[j] then
                swap(A, i, j)
            end if
        end for
    end for
    return A
end function
```

```
function BubbleSort(A : array)
    let n be the size of A
    repeat ← true
    while repeat do
        repeat ← false
        for i from 0 to n − 2 do
            if A[i] > A[i + 1] then
                swap(A, i, i + 1)
                repeat ← true
            end if
        end for
    end while
    return A
end function
```

Question 1

The closed-form runtime expression $T(n)$ for the number of compares between array items executed by ExchangeSort is:

A. $T(n) = \frac{1}{2}n^2 - \frac{1}{2}n$

B. $T(n) = \frac{1}{2}n^2 + \frac{1}{2}n$

C. $T(n) = n^2 - 1$

D. $T(n) = n^2 + 1$

E. $T(n) = n^2$

Question 2

The closed-form runtime expression $T(n)$ for the maximum number of Swap calls made by ExchangeSort is:

A. $T(n) = \frac{1}{2}n^2 - \frac{1}{2}n$

B. $T(n) = \frac{1}{2}n^2 + \frac{1}{2}n$

C. $T(n) = n^2 - 1$

D. $T(n) = n^2 + 1$

E. $T(n) = n^2$

Question 3

The closed-form runtime expression $T(n)$ for the maximum number of Swap calls made by BubbleSort is:

A. $T(n) = \frac{1}{2}n^2 - \frac{1}{2}n$

B. $T(n) = \frac{1}{2}n^2 + \frac{1}{2}n$

C. $T(n) = n^2 - 1$
D. $T(n) = n^2 + 1$
E. $T(n) = n^2$

Question 4

If $f(n) \in \Omega(g(n))$, then $4^{f(n)} \in \Omega\left(4^{g(n)}\right)$.

A. Necessarily True

B. Possibly True

C. Necessarily False

The ARRANGE function below rearranges the input array A content so that positive numbers are in even indices and negative numbers are in odd indices. The correctness of the algorithm requires that A be an array of size n, where n is an even integer. Also, A must contain $n/2$ positive integers and $n/2$ negative integers.

```
1:  function ARRANGE(A : array)
2:      let n be the size of A
3:      for i from 0 to n − 1 do
4:          if ¬((i is even ∧ A[i] > 0) ∨ (i is odd ∧ A[i] < 0)) then
5:              j ← i + 1
6:              repeat ← true
7:              while repeat ∧ j < n do
8:                  if (i is even ∧ A[j] > 0) ∨ (i is odd ∧ A[j] < 0) then
9:                      SWAP(A, i, j)
10:                     repeat ← false
11:                 end if
12:                 j ← j + 1
13:             end while
14:         end if
15:     end for
16:     return A
17: end function
```

Question 5

The following are arrays of size $n = 6$. Which input array will make ARRANGE to call SWAP $n/2$ times?

A. $[1, -2, 3, 4, -5, -6]$

B. $[1, 2, 3, -4, -5, -6]$

C. $[-1, 2, -3, 4, -5, 6]$

D. $[-1, -2, 3, 4, 5, -6]$

E. $[-1, -2, 3, 4, -5, 6]$

Question 6

Let K be an array of size n where the first $n/2$ items are negative, and the last $n/2$ items are positive. For instance, $K = [-1, -2, -3, 4, 5, 6]$ with $n = 6$. The runtime expression $T(n)$ for the number of times the condition in line 8 of the algorithm is evaluated when we call ARRANGE(K) is:

A. $T(n) = \frac{1}{8}n^2 - \frac{1}{4}n$

B. $T(n) = \frac{1}{8}n^2 + \frac{1}{4}n$

C. $T(n) = \frac{1}{4}n^2 + \frac{1}{2}n$

D. $T(n) = \frac{1}{2}n^2 - \frac{1}{2}n$

E. $T(n) = \frac{1}{2}n - \frac{1}{2}$

Consider a linked list where each node has an item and a pointer to the next node. A linked list has a cycle if at least one node exists in the list whose next pointer eventually points back to a previous node, forming a closed loop. The following is a cycle detection algorithm commonly known as the Tortoise and Hare algorithm:

```
function HasCycle(head : node)
    slow ← head
    fast ← head
    while fast ≠ null ∧ fast.next ≠ null do
        slow ← slow.next
        fast ← fast.next.next
        if slow = fast then
            return true
        end if
    end while
    return false
end function
```

Question 7

Suppose you run this function on a list with n nodes, with a cycle starting at node k (where $1 \leq k \leq n$). What can be said about the number of steps the *slow* and *fast* pointers take before they meet?

A. The *slow* and *fast* pointers will meet exactly after n steps.

B. The *slow* pointer will always take n steps before meeting the *fast* pointer, regardless of where the cycle starts.

C. The *slow* pointer will always enter the cycle before the *fast* pointer.

D. The number of steps taken by the *slow* pointer before they meet depends on both the length of the list before the cycle and the length of the cycle itself.

E. The *slow* and *fast* pointers may never meet if the cycle starts too close to the head of the list.

Question 8

Let Q in the following pseudocode be a fixed-capacity queue implemented with a circular array (i.e., the elements wrap around to reuse space at the front, as discussed in class).

```
function UnnecessaryAlgorithm(n : Z+)
    Let Q be a queue implemented in an array of capacity n
    for i from 0 to 4 do
        Q.enqueue(i)
    end for
    for i from 0 to 2 do
        Q.dequeue()
    end for
    for i from 5 to 11 do
        Q.enqueue(i)
    end for
    Q.dequeue()
    Q.enqueue(15)
end function
```

At which index in the array is element 15 after calling UnnecessaryAlgorithm(10)?

A. 2

B. 4

C. 6

D. 8

E. 0

Let T be the binary tree for which the following traversals hold:

- Preorder: T U T S R C E U R S
- Inorder: S T R U C T U R E S
- Postorder: S R T C U R U S E T

Question 9

The height of T is:

A. 2

B. 3

C. 4

D. 5

E. 6

Question 10

T is:

A. A balanced binary tree

B. A complete binary tree

C. A min binary heap

D. A max binary heap

E. A full binary tree

Question 11

The most populated level of T is:

A. 0

B. 1

C. 2

D. 3

E. 4

Question 12

Let B be the binary tree representation of a max binary heap. The values $[8, 11, 3, 10, 7, 4, 6]$ are inserted in B (one at that time in the given order). What is the inorder traversal of B?

A. 10, 11, 8, 7, 6, 4, 3

B. 8, 10, 7, 11, 3, 6, 4

C. 8, 10, 11, 7, 3, 6, 4

D. 8, 10, 7, 11, 4, 6, 3

E. 3, 4, 6, 7, 8, 10, 11

Question 13

Let B' be the binary tree representation of a max binary heap built using Heapify($[8, 11, 3, 10, 7, 4, 6]$). What is the inorder traversal of B'?

A. 10, 11, 8, 7, 6, 4, 3

B. 8, 10, 7, 11, 3, 6, 4

C. 8, 10, 11, 7, 3, 6, 4

D. 8, 10, 7, 11, 4, 6, 3

E. 3, 4, 6, 7, 8, 10, 11