Week 03 Quiz

Plagiarism declaration

By submitting work for this quiz I hereby declare that I understand the University's policy on **academic integrity (https://academicintegrity.unimelb.edu.au/)** and that the work submitted is original and solely my work, and that I have not been assisted by any other person (collusion) apart from where the submitted work is for a designated collaborative task, in which case the individual contributions are indicated. I also declare that I have not used any sources without proper acknowledgment (plagiarism). Where the submitted work is a computer program or code, I further declare that any copied code is declared in comments identifying the source at the start of the program or in a header file, that comments inline identify the start and end of the copied code, and that any modifications to code sources elsewhere are commented upon as to the nature of the modification.

Due Mar 26 at 23:59 **Points** 10 **Questions** 10
Available Mar 17 at 10:00 - Mar 26 at 23:59 10 days **Time Limit** None
Allowed Attempts Unlimited

This quiz was locked Mar 26 at 23:59.

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	101 minutes	9 out of 10

Score for this attempt: **9** out of 10
Submitted Mar 25 at 5:57
This attempt took 101 minutes.

Question 1 — 0 / 1 pts

A sorted array supports fast search (using binary search) but slow insertion when we must retain sorted order.

orrect Answer

ou Answered

- True
- False

Given a sorted array, binary search looks the the middle element in the array and compares it to the item we are looking for. If this element is the item we seek, we are done. If not, binary search splits the array into two parts and determines whether the item we seek is in the first or second half. If our item is smaller than the middle element, we know that it must be in the first half of the array. We can immediately discard one half of the array from consideration. This process continues until we either find the required item, or we can prove that it is not in the array. In an unsorted array of n items, however, we must consider each element in turn, requiring up to n comparisons. Binary search, in contrast, makes O(log2n) comparisons in the worst case. Inserting an element into the correct position in a sorted array may require moving a large number of elements one position to the right (and potentially more memory to be allocated to the array).

Question 2

1 / 1 pts

Given an unsorted singly-linked list L of n items, which one of the following statements is **False.** Note that a singly-linked list is a linked list in which each node contains a single pointer to the next node in sequence.

Correct!

- Insertion and deletion of an item I are both O(1) operations (worst case).

While insertion of an item into an unsorted singly-linked list is an O(1) operation in the worst case, deletion of an item is not. Deletion has a worst case complexity of O(n). To delete an item I from L, assuming we have a pointer to I, we also need to have a pointer to the node just before I in L. This node must then be altered to point to the node just after I in L. In a singly-linked list, each node does not maintain a pointer to its immediate predecessor. Consequently, we have to step through the list to find this predecessor. To insert a node into an unsorted linked list, we can simply insert it at the beginning!

- Search for a specific item I is an O(n) operation (worst case).
- Finding the smallest or largest item in L are both O(n) operations (worst case).

Question 3

1 / 1 pts

Consider this instance of the Assignment Problem (introduced in tutorial exercises).

	Job 1	Job 2	Job 3	Job 4
Contractor 1	13	16	12	11
Contractor 2	15	17	12	12
Contractor 3	14	14	13	13
Contractor 4	13	10	10	11

Match each contractor to a job so as to minimise the cost.

Correct!

Contractor 1 Job 4

Correct!

Contractor 2 Job 3

Correct!

Contractor 3 Job 1

Correct!

Contractor 4 Job 2

Yes, that's right.

Question 4

1 / 1 pts

This function uses a curious mix of iteration and recursion:

```
function F(n)
    if n < 1
        return 1
    t <- 0
    for i <- 0 to n
        for j <- i to n
            t <- t + j
    return t + F(n-1)
```

The number of basic operations (additions and subtractions) performed is:

- $\Theta(n)$
- $\Theta(n^2)$
- $\Theta(n^2 \log n)$

Correct!

- $\Theta(n^3)$
- $\Theta(n^4)$

Well done.

Question 5

1 / 1 pts

Given two functions $f(n) = n^n$ and $g(n) = n!$, which of the following statement is correct?

Correct!

- (•) f(n) has a higher growth rate than g(n)
- ( ) g(n) has a higher growth rate than f(n)
- ( ) They have same growth rate
- ( ) It's hard to compare their growth rates without knowing n

Question 6

1 / 1 pts

In Lecture 5 we discussed the brute-force approach to string search. How many character comparisons will the algorithm make when searching for 'lido' in the string

'supercalifragilisticexpialidocious' ?

Correct!

33

orrect Answers

33 (with margin: 0)

That's right - well done.

Question 7

1 / 1 pts

One of my books has 589 pages, numbered consecutively. Every page has a page number, the first being 1. How many decimal digits were used to type the 589 page numbers?

- 1657
- 1659

Correct!

- 1660
- 1661
- 1667

Right - there are 9 page numbers of length 1, 90 of length 2, and 490 of length 3.

Question 8

1 / 1 pts

In Lecture 6 we gave a recursive algorithm for solving the Tower of Hanoi puzzle. Assume we have a tower of 24 disks to move, and each move (moving one disk from one peg to another) takes one minute. The total time taken will be:

- Approximately six days
- Approximately two months
- Approximately one year

Approximately ten years

Correct!

Approximately 32 years

Yes, that's correct. The original puzzle, by Edouard Lucas, asked how long it would take to move 64 disks, not 24. Now try to estimate that. Hint: It will take longer than the estimated age of the universe!

Question 9

1 / 1 pts

What does the following recursive function do? You may assume that the input n is always greater than 10.

```
function f(n)
   if n >= 2 than
      return f(n-2) * n
   return 2
```

It calculates the product of all odd or even integers between 1 and n

It calculates the sum of integers between 2 and n

Correct!

It calculates 2 times the product of all odd or even integers between 1 and n

It calculates the factorial of n

Question 10

1 / 1 pts

Which one best describes the complexity (number of basic operation executions) of the following function?

```
function f(A[0..n-1], k)
    if n < 1 then
        return 0
    if A[n-1] == k then
        return 1 + f(A[0..n-2], k)
    return f(A[0..n-2], k)
```

- $O(n)$
- **Correct!** $\Theta(n)$
- $\Omega(n^2)$
- $O(\sqrt{n})$

Quiz Score: **9** out of 10