Lecture 11
Semester 1 Revision

COMP26120

Giles Reger and Lucas Cordeiro

December 2019

Headlines

January Exam is 1.5 hours (90 minutes)

The Exam is Online Only via Blackboard

C programming is not examinable

The two main topics this semester have been

- Complexity Analysis and Divide-and-Conquer [3 lectures]
- Data Structures [3 lectures]

Reminder: Course Assessment

Coursework (50%):

- 25% each semester
- In each semester
  - 20% from coding exercises
  - 5% from 3 online quizzes (see website for when)
  - Quizzes provide early feedback on conceptual issues

Exam (50%):

- 15% first semester, 35% second semester
- Material (general concepts) covered in labs is examinable
- Note: new staff means exams likely to be a little different in style

Reminder: Course Assessment

Coursework (50%): ← note split

- 25% each semester
- In each semester
  - 20% from coding exercises
  - 5% from 3 online quizzes (see website for when)
  - Quizzes provide early feedback on conceptual issues

Exam (50%): ← note split

- 15% first semester, 35% second semester
- Material (general concepts) covered in labs is examinable
- Note: new staff means exams likely to be a little different in style

Reminder: Health Warning 1

Warning 1

This course unit is rather different from others. It is important that you take the following into account:

- You will be expected to read a considerable amount of material outside the lectures - this includes the course textbook and other material too.
- Lectures will give some guidance about course content and topics, laboratory exercises, etc., but much of the material of the course unit is not lectured!

Examinable Topics

C Programming is assessed in labs only.

Complexity Analysis and Divide-and-Conquer

- Divide and Conquer paradigm
- Sorting Algorithms
- Algorithmic Complexity
- Sums and Integrals
- Proof by Induction

Data Structures

- Abstract Datatypes
- Binary trees
- AVL trees
- Hashing structures
- Skip Lists

Relation to Previous Papers

Computing the complexity of recursive programs (and the required mathematics) is included in this year's syllabus and first appeared last year. .

With respect to past papers:

- We moved data structure material from semester 2 to semester 1 last year
- Relevant to this semester would be (e.g.) question 3 of paper 2 from 2015-2017 and question 4 from paper 2 in 2015
- Some parts of other questions may be relevant - you can use your knowledge of the relevant topics from this semester to make a judgement

Complexity Question 1

What is the most accurate (e.g. smallest correct) time complexity of this (C-like) pseudocode?

```
for ( i=1; i < n; i *= 2 ){
  for ( j = n; j > 0; j /= 2 ) {
    for ( k = j; k < n; k += 2 ) {
      sum += (i + j * k );
    }
  }
}
```

Options:

- $O(n\,(\log n)^2)$
- $O(n^2)$
- $O(n\,(\log n))$
- None of the above.

Complexity Question 1

What is the most accurate (e.g. smallest correct) time complexity of this (C-like) pseudocode?

```
for ( i=1; i < n; i *= 2 ){
  for ( j = n; j > 0; j /= 2 ) {
    for ( k = j; k < n; k += 2 ) {
      sum += (i + j * k );
    }
  }
}
```

Options:

- $O(n\ (\log\ n)^2)$ - Three nested loops. Inner is $O(n)$, others are $O(\log\ n)$
- $O(n^2)$
- $O(n\ (\log\ n))$
- None of the above.

Complexity Question 2

Consider the scenario

> *John is working as a software developer at BoostCode UK Ltd. He needs to compare two implementations of sorting algorithms on the same machine, which will be later integrated into a software product. In particular, for inputs defined by size n, the first algorithm denoted by A runs in* $4n^2$*, while the second algorithm denoted by B runs in* $128n\log_2 n$*. John needs to compute values of n where algorithm A beats algorithm B so that he can make the best choice of which one will be integrated into the product. For which values of n will algorithm A run faster than algorithm B?*

Options:

- 256
- 128
- 64
- None of the Above

Complexity Question 2

Consider the scenario

> *John is working as a software developer at BoostCode UK Ltd. He needs to compare two implementations of sorting algorithms on the same machine, which will be later integrated into a software product. In particular, for inputs defined by size n, the first algorithm denoted by A runs in $4n^2$, while the second algorithm denoted by B runs in $128n \log_2 n$. John needs to compute values of n where algorithm A beats algorithm B so that he can make the best choice of which one will be integrated into the product. For which values of n will algorithm A run faster than algorithm B?*

Options:

- 256
- 128
- 64
- None of the Above

Complexity Question 3

Solve the recurrence equation

$$T(n) = T\left(\frac{2n}{3}\right) + 1$$

using the master method to give tight asymptotic bounds.

Options:

- $T(n) = \theta(\log\ n)$
- $T(n) = \theta(n)$
- $T(n) = \theta(1)$
- None of the Above

Other questions might include:

- Which Case applies when applying the master method?
- Identify the parts $a, b, f(n)$

Complexity Question 3

Solve the recurrence equation

$$T(n) = T\left(\frac{2n}{3}\right) + 1$$

using the master method to give tight asymptotic bounds.

Options:

- $T(n) = \theta(\log\ n)$ - Case 2 applies
- $T(n) = \theta(n)$
- $T(n) = \theta(1)$
- None of the Above

Other questions might include:

- Which Case applies when applying the master method?
- Identify the parts $a, b, f(n)$

Complexity Question 4

Which sentences about sums are correct?

1. The perturbation method typically works for sums that do not contain exponentials
2. The sum $\Sigma_{i=1}^{n} a$ can be approximated by integrals to $O(n)$
3. The sum $\Sigma_{i=1}^{n} i^3$ can be approximated by integrals to $O(n^4)$
4. The sum $\Sigma_{i=1}^{n} \frac{1}{i}$ can be approximated by integrals to $O(\ln\ n)$

Complexity Question 4

Which sentences about sums are correct?

1. The perturbation method typically works for sums that do not contain exponentials
2. The sum $\Sigma_{i=1}^{n} a$ can be approximated by integrals to $O(n)$
3. The sum $\Sigma_{i=1}^{n} i^3$ can be approximated by integrals to $O(n^4)$
4. The sum $\Sigma_{i=1}^{n} \frac{1}{i}$ can be approximated by integrals to $O(\ln\ n)$

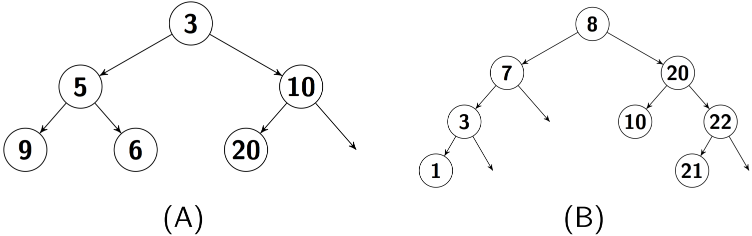
b. $\displaystyle\sum_{i=1}^{n} a = \int_1^n a = O(n).$

c. $\displaystyle\sum_{i=1}^{n} i^3 = \int_1^n i^3 = O(n^4).$

d. $\displaystyle\sum_{i=1}^{n} 1/i = \int_1^n 1/i = O(\ln\ n).$

Data Structures Question 1

Consider the following two trees

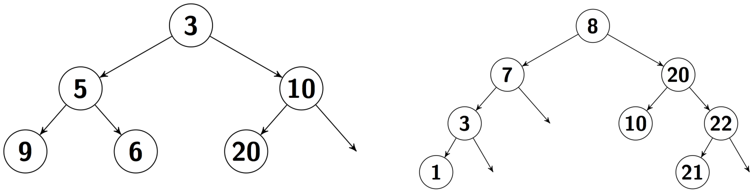


Which statements are TRUE:

- A is ordered
- B is ordered
- A is complete
- B is complete
- A has the Heap-Order property
- B has the Height-Balance property
- The balance of node 7 in A is -2
- The balance of node 8 in A is -2

Data Structures Question 1

Consider the following two trees



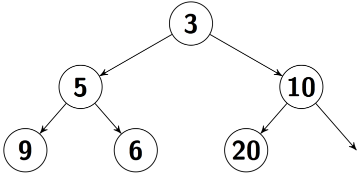
(A) (B)

Which statements are TRUE:

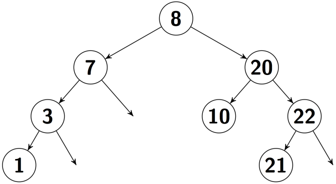
- A is ordered
- B is ordered
- A is complete
- B is complete
- A has the Heap-Order property
- B has the Height-Balance property
- The balance of node 7 in A is -2
- The balance of node 8 in A is -2

Data Structures Question 1

Consider the following two trees



(A)



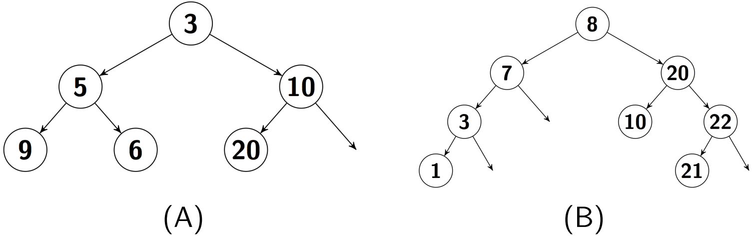
(B)

Which statements are TRUE:

- A is ordered
- B is ordered
- A is complete
- B is complete
- A has the Heap-Order property
- B has the Height-Balance property
- The balance of node 7 in A is -2
- The balance of node 8 in A is -2

Data Structures Question 1

Consider the following two trees

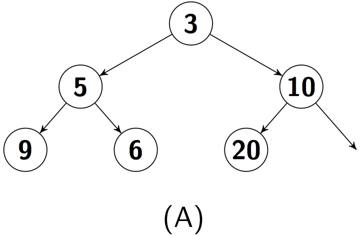


Which statements are TRUE:

- A is ordered
- B is ordered
- A is complete
- B is complete
- A has the Heap-Order property
- B has the Height-Balance property
- The balance of node 7 in A is -2
- The balance of node 8 in A is -2

Data Structures Question 1

Consider the following two trees



Which statements are TRUE:

- A is ordered
- B is ordered
- A is complete
- B is complete
- A has the Heap-Order property
- B has the Height-Balance property
- The balance of node 7 in A is -2
- The balance of node 8 in A is -2

Data Structures Question 2

Consider the following scenario

> *I am creating an application that generates a Word of the Day. I want the words to get more interesting over time so I've associated an interestingness score with each word. I have the list in alphabetical order. At the start of the year I insert the words. Each day I return a new word.*

Which ADT do I need?

- Set
- Stack
- Dictionary
- Priority Queue

Which Data structure should I use?

- Dynamic Array
- AVL tree
- Skip List
- Binary Heap

Data Structures Question 2

Consider the following scenario

> *I am creating an application that generates a Word of the Day. I want the words to get more interesting over time so I've associated an interestingness score with each word. I have the list in alphabetical order. At the start of the year I insert the words. Each day I return a new word.*

Which ADT do I need?

- Set
- Stack
- Dictionary
- Priority Queue

Which Data structure should I use?

- Dynamic Array
- AVL tree
- Skip List
- Binary Heap

Data Structures Question 2

Consider the following scenario

> *I am creating an application that generates a Word of the Day. I want the words to get more interesting over time so I've associated an interestingness score with each word. I have the list in alphabetical order. At the start of the year I insert the words. Each day I return a new word.*

Which ADT do I need?

- Set
- Stack
- Dictionary
- Priority Queue

Which Data structure should I use?

- Dynamic Array
- AVL tree
- Skip List
- Binary Heap

Data Structures Question 3

Assertion: Inserting 1, 10, 13, 23, 29 into a hash table of size 100 using the hash function $h(n) = n \bmod 9$ and linear probing would lead to 2 collisions.

BECAUSE

Reason: Quadratic Probing uses the formula $A[(i + j^2) \bmod N]$ to avoid clustering effects

- The assertion and reason are both correct and the reason is valid.
- The assertion and reason are both correct but the reason is invalid.
- The assertion is correct, but the reason is incorrect.
- The assertion is incorrect, but the reason is correct.
- Both the answer and the reason are incorrect.

Data Structures Question 3

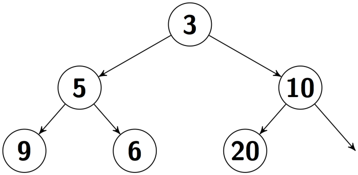
Assertion: Inserting 1, 10, 13, 23, 29 into a hash table of size 100 using the hash function $h(n) = n \bmod 9$ and linear probing would lead to 2 collisions.

BECAUSE

Reason: Quadratic Probing uses the formula $A[(i + j^2) \bmod N]$ to avoid clustering effects

- The assertion and reason are both correct and the reason is valid.
- The assertion and reason are both correct but the reason is invalid.
- The assertion is correct, but the reason is incorrect.
- The assertion is incorrect, but the reason is correct.
- Both the answer and the reason are incorrect.

Consider the following Binary Heap

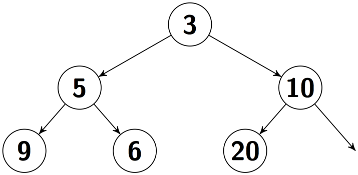


How many popMin operations are required before the tree has height 1?

How many popMin operations are required before 20 is the left of the root node?

Data Structures Question 4

Consider the following Binary Heap

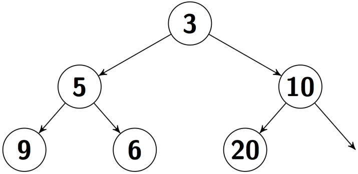


How many popMin operations are required before the tree has height 1? 3

How many popMin operations are required before 20 is the left of the root node?

Data Structures Question 4

Consider the following Binary Heap



How many popMin operations are required before the tree has height 1? 3

How many popMin operations are required before 20 is the left of the root node? 1

Data Structures Questions in General

The kinds of questions I might ask (similar to quiz):

- To identify the correct definition for a term
- To identify where a data structure has a property
- To identify whether a data structure is the result of certain operations e.g. inserting certain numbers into an hash table, AVL tree or binary heap, or removing a node from an AVL tree and re-balancing

Summary

Good Luck!

Any Questions?