Session 1 - AUT/BAU 2019 - Software Main Exam

UTS

SEAT NUMBER:

STUDENT NUMBER:

SURNAME:
(FAMILY NAME)

OTHER NAMES:

This paper and all materials issued must be returned at the end of the examination. They are not to be removed from the exam centre.

Examination Conditions:

It is your responsibility to fill out and complete your details in the space provided on all the examination material provided to you. Use the time before your examination to do so as you will not be allowed any extra time once the exam has ended.

You are **not** permitted to have on your desk or on your person any unauthorised material. This includes but not limited to:

- Mobile phones
- Smart watches and bands
- Electronic devices
- Draft paper (unless provided)
- Textbooks (unless specified)
- Notes (unless specified)

You are **not** permitted to obtain assistance by improper means or ask for help from or give help to any other person.

If you wish to **leave and be re-admitted** (including to use the toilet), you have to wait until **90 mins** has elapsed.

If you wish to **leave the exam room permanently**, you have to wait until **60 mins** has elapsed.

You are not permitted to leave your seat (including to use the toilet) during the final 15 mins.

During the examination **you must first seek permission** (by raising your hand) from a supervisor before:

- Leaving early
- Using the toilet
- Accessing your bag

Misconduct action will be taken against you if you breach university rules.

Declaration: I declare that I have read the advice above on examination conduct and listened to the examination supervisor's instructions for this exam. In addition, I am aware of the university's rules regarding misconduct during examinations. I am not in possession of, nor do I have access to, any unauthorised material during this examination. I agree to be bound by the university's rules, codes of conduct, and other policies relating to examinations.

Signature:

Date:

31251 Data Structures and Algorithms

Time Allowed: 120 minutes.

Reading time: 10 minutes.

Reading time is for reading only. You are not permitted to write, calculate or mark your paper in any way during reading time.

Open Book
No Calculators Permitted

Permitted materials for this exam:

Any Notes
Any Textbook
Any paper materials that are handwritten, photocopied, or typed
Drawing Instruments

Materials provided for this exam:

1 x 8 Page Booklet
1 x General Purpose Answer Sheets (GPAS-240R)

Students please note:

All code fragments are in C++ using the C++17 standard (as used in the subject).

This exam has two parts:

Part A (20 marks) consists of 20 multiple choice questions. Please answer these on the multiple choice answer sheet provided.

Part B (30 marks) consists of 4 short answer questions. Please answer these in the booklet provided.

Do not open your exam paper until instructed.

Rough work space
Do **not** write your answers on this page.

Part A: Multiple Choice Questions

Question 1. [1 mark]

Consider the following class:

```
class node {
private:
  int data;
  node * next;
public:
  node(int data, node* next){
    this->data = data;
    this->next = next;
  }

  int get_data() const { return this->data; }

  bool p(const int& d) const {
    if (data == d) return true;
    if (next == nullptr) return false;
    return next->p(d);
  }
};
```

and the following code snippet:

```
node * x = new node(5, nullptr);
node * y = new node(3, x);
y->p(5);
```

What is the result of the code snippet?

A. `p` is executed on both nodes and returns `true`.
B. `p` is executed on both nodes and returns `false`.
C. `p` is executed only on the first node and returns `true`.
D. `p` is executed only on the first node node and returns `true`.

Question 2. [1 mark]

What will the following code snippet print out?

```
int i = 10;
int * p = &i;
p += 3;
std::cout << *p << std::endl;
```

A. 13.
B. The address of `i`.
C. 10.
D. An unknown value from memory.

Question 3. [1 mark]

Which of the following code fragments correctly creates an array called `a` of `int`s of size 5?

A. `int[] a = new int[5];`
B. `int a(5);`
C. `int a[5];`
D. `int a = [5];`

Question 4. [1 mark]

Which of the following code fragments correctly creates a reference to an `int` called `p` and points it at an `int` called `a`?

A. `int * p = &a;`
B. `int * p = a;`
C. `int & p = &a;`
D. `int & p = a;`

Question 5. [1 mark]

Consider the following two methods, "mystery" and "main":

```
#include <iostream>
#include <vector>

int mystery(std::vector<int> x1, std::vector<int> x2) {
  int i1 = 0; // indicates a position in array x1
  int i2 = 0; // indicates a position in array x2

  int count = 0;

  while ((i1 < x1.length) || (i2 < x2.length)) {
    if (i1 == x1.length) {
      ++count;
      ++i2;
    }
    else if (i2 == x2.length) {
      ++count;
      ++i1;
    }
    else if (x1[i1] == x2[i2]) {
      ++i1;
      ++i2;
    }
    else if (x1[i1] < x2[i2]) {
      ++count;
      ++i1;
    }
    else {
      ++count;
      ++i2;
    }
  }
  return count;
}

int main() {
  std::vector<int> x1 = {1, 3};
  std::vector<int> x2 = {1, 2, 5};
  std::cout << mystery(x1,x2)) << std::endl;
}
```


If the method "main " is executed, what value is outputted?

A. 1
B. 2
C. 3
D. 4

Question 6. [1 mark]

Given a stack with the standard push and pop functions, what value is on the top of the stack of the following sequence of operations: `push(6)`, `push(1)`, `push(3)`, `pop()`, `push(4)`, `pop()`, `pop()`, `pop()`, `push(5)`, `pop()`?

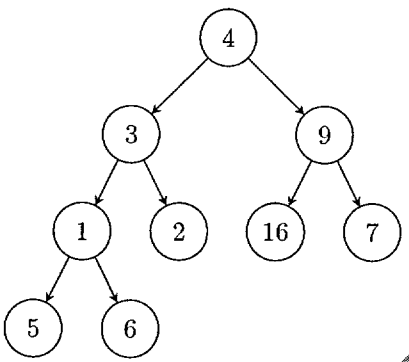
A. 5.
B. 1.
C. 6.
D. The stack is empty.

Question 7. [1 mark]

Given a standard linked list what is the asymptotic (big-oh) time for finding the item in postion $\frac{n}{2}$ out of n elements?

A. $O(1)$
B. $O(\log_2(n))$
C. $O(n)$
D. $O(n^2)$

Questions 8-11 refer to the following tree:



Question 8. [1 mark]

What is is the visiting order of a depth-first pre-order traversal of the tree?

A. 5, 1, 6, 3, 2, 4, 16, 9, 7
B. 4, 3, 1, 5, 6, 2, 9, 16, 7
C. 4, 3, 1, 5, 9, 16, 2, 6, 7
D. 4, 3, 9, 1, 2, 16, 7, 5, 6

Question 9. [1 mark]

What is is the visiting order of a depth-first in-order traversal of the tree?

A. 5, 6, 1, 2, 3, 16, 7, 9, 4
B. 5, 1, 6, 3, 2, 4, 16, 9, 7
C. 5, 6, 1, 2, 3, 16, 7, 9, 4
D. 4, 3, 1, 5, 6, 2, 9, 16, 7

Question 10. [1 mark]

What is is the visiting order of a depth-first post-order traversal of the tree?

A. 4, 3, 1, 5, 6, 2, 9, 16, 7
B. 5, 1, 6, 3, 2, 4, 16, 9, 7
C. 5, 6, 2, 16, 7, 1, 3, 9, 4
D. 5, 6, 1, 2, 3, 16, 7, 9, 4

Question 11. [1 mark]

What is is the visiting order of a breadth-first traversal of the tree?

A. 4, 3, 1, 5, 6, 2, 9, 16, 7
B. 5, 1, 6, 3, 2, 4, 16, 9, 7
C. 5, 6, 1, 2, 3, 16, 7, 9, 4
D. 4, 3, 9, 1, 2, 16, 7, 5, 6

Question 12. [1 mark]

Which of the following statements is **TRUE**?

A. $4n^2 - 16n \log n + 2 \in O(n)$
B. $56n + 3n^2 + n \log n \in \Omega(n)$
C. $2n^3 + 16n^2 + 2n^4 \in \Theta(n^3)$
D. $4n - 3n + 2 \in O(4)$

Question 13. [1 mark]

Consider the following observations of two pieces of code:

- Implementation $I_{\mathcal{A}}$ of an algorithm $\mathcal{A}$ solves all problems of size at most 4000 elements with running time $T_{\mathcal{A}}$ less than 10 seconds.
- Implementation $I_{\mathcal{B}}$ of algorithm $\mathcal{B}$ has running time $T_{\mathcal{B}}$ of up to a minute on some instances of size at most 4000.

Which of the following statements can be mathematically concluded from this information?

A. $T_{\mathcal{A}} \in O(T_{\mathcal{B}})$
B. $T_{\mathcal{B}} \in O(T_{\mathcal{A}})$
C. $T_{\mathcal{A}} \in \Theta(T_{\mathcal{B}})$
D. Nothing can be concluded from this information.

Question 14. [1 mark]

Which one of the following statements is **FALSE**?

A. $n \log n \in O(n^{\frac{3}{2}})$
B. $2^{n+1} \in O(2^n)$
C. $\sqrt{n} \in O(\log n)$
D. $n^2 \in \Theta(n^2)$

Question 15. [1 mark]

What is the worst-case asymptotic (big-oh) running time of inserting a new element into a Binary Search Tree with n vertices?

A. $O(1)$
B. $O(\log n)$
C. $O(n)$
D. $O(n \log n)$

Question 16. [1 mark]

What is the worst-case asymptotic (big-oh) running time of inserting a new element at the start of a Linked List?

A. $O(1)$
B. $O(\log n)$
C. $O(n)$
D. $O(n \log n)$

Question 17. [1 mark]

Assuming there are no collisions, what is the expected asymptotic (big-oh) running time of inserting a new element into a Hashmap which already has n elements?

A. $O(1)$
B. $O(\log n)$
C. $O(n)$
D. $O(n \log n)$

Question 18. [1 mark]

Which algorithmic paradigm is Quicksort an example of?

A. Greedy
B. Dynamic Programming
C. Divide and Conquer
D. Complete Enumeration

Question 19. [1 mark]

Which algorithmic paradigm is Prim's Algorithm an example of?

A. Greedy
B. Dynamic Programming
C. Divide and Conquer
D. Complete Enumeration

Question 20. [1 mark]

Which of the following is true of Open Addressing/Closed Hashing?

A. It requires no additional storage.
B. It requires additional storage.
C. It requires auxilliary data structures.
D. It handles removal of elements easily.

Part B: Short Answer Questions

Question 21. [9 marks]

This question concerns heaps, their properties and their implementations.

i. **[2 marks]** What is the defining property of a min-heap?
ii. **[2 marks]** What is the tight asymptotic (big-oh) running time of inserting a new element into a heap with n elements? Justify this running time.
iii. **[3 marks]** Beginning with an empty max-heap, insert the following elements in the given order: 5, 6, 2, 8, 1, 3, 7, 10, 4. Sketch the heap after every insertion is completed (you don't need to sketch the full insertion process).
iv. **[2 marks]** Starting with your answer to part iii of this question, sketch the full process of removing the largest element from the heap, and re-heapifying.

Question 22. [6 marks]

The following questions are short, "explain in plain English" questions. You will be given some code and you are required to described the purpose of it. For example, given the following code:

```
if (a < b)
    std::cout << a << std::endl;
else
    std::cout << b << std::endl;
```

a good answer would be "It prints out the smaller of the two values held by variables a and b."
Do not give a line by line description of the code. The code should also not be evaluated on whether it compiles or not, your task is to determine the *intent* of the code, not the ability of the "programmer".

i. **[1 mark]** Consider the following code:

```
class node {
private:
    int data;
    node* next;
public:

  node(int data, node* next){
      this->data = data;
      this->next = next;
  }

  int get_data() const { return this->data; }

  bool p(const int & d) const {
      if (data == d) return true;
      if (next == nullptr) return false;
      return next->p(d);
  }
};
```

Assume you have a linked list constructed of such nodes. What property does calling the function p on the head of the linked list compute?

ii. **[1 mark]** In one sentence, explain in plain English what the following function does:

```
int w(const std::vector<int>& v) const {
  int n = 0;
  for (std::vector<int>::iterator itr = v.begin(); itr < v.end(); ++itr){
    if (*itr > 0) n += *itr;
  }
  return n;
}
```

iii. **[1 mark]** In one sentence, explain in plain English what the following method does:

```
template <typename T>
std::list<T> x(std::list<T> v) {
  for (auto itr = v.begin(); itr != v.end(); ++itr) {
    auto z = itr;
    for (auto itr2 = itr; itr2 != v.end(); ++itr2) {
      if (*itr2 < *z) z = itr2;
    }
    auto y = *z;
    *z = *itr;
    *itr = y;
  }
  return v;
}
```

iv. [1 mark] In one sentence, explain in plain English what the method v does:

```
struct node {
  int data;
  node * next;
}

void v(node * x) {
  node * y = x;
  while (y != nullptr) {
   y->data++;
   y = y->next;
  }
}
```

v. [1 mark] In one sentence, explain in plain English what the method z does:

```
struct node {
  int data;
  node * next;
}

int z(node * v, int x){
  if (v == nullptr) return 0;
  if (x > 0) return v->data + z(v->next, x * -1);
  return z(v->next, x * -1);
}
```

vi. [1 mark] In one sentence, explain in plain English what the method `mystery` in Question 6 does.

Question 23. [6 marks]

The following questions require you to *implement* a simple algorithm. You should write in C++, however you will not be marked down for minor syntax errors. The goal is to correctly implement the *logic* of the algorithm. Given the following skeleton:

```
struct node {
  int data;
  node * next;
}

int unique_count(node * x, node * y) {
  if (x == nullptr && y == nullptr)
    return ***line 0***

  if (y == nullptr)
    return ***line 1***

  if (x == nullptr)
    return ***line 2***

  if (x->data == y->data)
    return ***line 3***

  if (x->data < y->data)
    return ***line 4***

  return ***line 5***
}
```

your task is to implement an algorithm that works on two, sorted, linked lists, and returns the number of elements that are in the list pointed to by x, but not in the list pointed to by y.

i. [1 mark] What code replaces `***line 0***`?

ii. [1 mark] What code replaces `***line 1***`?

iii. [1 mark] What code replaces `***line 2***`?

iv. [1 mark] What code replaces `***line 3***`?

v. [1 mark] What code replaces `***line 4***`?

vi. [1 mark] What code replaces `***line 5***`?

Question 24. [9 marks]

i. **[1 mark]** In plain English, what is the traversal rule that Binary Search trees use to find where to insert a new element into the tree, and to find if a value is in the tree?

ii. **[5 marks]** Starting with an empty Binary Search Tree, insert the following values into the tree in the order presented: 6, 12, 3, 1, 5, 4, 8, 9. Draw the tree after each insertion.

iii. **[2 marks]** Starting with the values 6, 12, 3, 1, 5, 4, 8, 9, what insertion order would produce the worst-case behaviour of the Binary Search Tree?

iv. **[1 mark]** What tree property to more advanced data structures like an AVL Tree, Red-Black Trees or 2,3-Trees try to maintain to avoid the worst case behaviour of the simple Binary Search Tree?