THE UNIVERSITY OF ADELAIDE
AUSTRALIA

Primary Examination, Semester 1, 2017

Algorithm Design and Data Structures
COMPSCI 1103, 2103

Official Reading Time:	10 mins
Writing Time:	120 mins
Total Duration:	130 mins

Questions	Time	Marks
Answer all 7 questions	120 mins	120 marks 120 Total

Instructions

- Begin each answer on a new page in the answer book.
- Examination material must not be removed from the examination room.

Materials

- Foreign language paper dictionaries permitted.

DO NOT COMMENCE WRITING UNTIL INSTRUCTED TO DO SO

Programming Fundamentals

Question 1

(a) You use the `new` keyword to allocate a dynamic variable and provide a pointer to that variable.

i. Where does the memory for this new variable come from?

[1 mark]

ii. Where is the pointer to the new variable stored?

[2 marks]

iii. Please explain why we do not have to manually delete local variables in C++, but have to do so for heap variables allocated using `new`.

[2 marks]

(b) Please consider each of the following statements carefully and give the answer **true** or **false** and justify your answer.

i. Linked lists are contiguous in memory.

[2 marks]

ii. The `virtual` keyword on the function means that you can now overload the function.

[2 marks]

(c) What is the output of the following code fragment?

```
int* x,y;
x = new int;
y = 15;
*x = 25;
cout << *x << " " << y << endl;
y = *x;
cout << *x << " " << y << endl;
*x = 50;
cout << *x << " " << y << endl;
```

[3 marks]

(d) "In C++, the vector template class provides bound checking." Is this statement true or false? Provide an explanation to support your answer.

[2 marks]

(e) Give an example of a brute-force strategy and where you might use it.

[4 marks]

[Total for Question 1: 18 marks]

Inheritance and Object Oriented Programming

Question 2

(a) The concept of polymorphism is associated with the mechanism known as *dynamic binding*. Please briefly explain what dynamic binding means.

[2 marks]

(b) What is a friend function?

[2 marks]

(c) Please clearly describe, in the context of C++, the difference between:

- overloading
- overriding

You may use diagrams where necessary.

[4 marks]

(d) Consider the following two classes.

```
class Pet
{
public:
    void print();  // Prints out the name of a Pet
    string name;
};
class Cat: public Pet
{
    void print();  // Prints out name and weight of a Cat.
    double weight;
};
```

i. What problem occurs when the following statements are executed? Note: the following operations are all legal.

```
Cat vcat;
Pet vpet;
vcat.name = "Bella";
vcat.weight = 3.8;
vpet = vcat;
```

[2 marks]

ii. Consider a different code fragment shown in the following.

```
Cat vcat;
Pet vpet;
vcat.name = "James";
vpet = vcat;
vpet.weight = 4.2;
```

Is the operation corresponding to the last statement legal? Briefly explain.

[2 marks]

iii. Please provide the modified class interfaces which make the following code block work as expected.

```
void Cat::print(){
    cout << "name: " << name << endl;
    cout << "weight: " << weight << endl;
}

Pet *ppet;
Cat *pcat;
pcat = new Cat;
ppet = pcat;
ppet -> print(); // Prints out the name and weight
```

[2 marks]

[Total for Question 2: 14 marks]

Recursion

Question 3

(a) What are the three requirements for successful recursion in C++?

[3 marks]

(b) Please explain the advantages and disadvantages of using recursion instead of an iterative approach.

[2 marks]

(c) i. Write a recursive function `int func(int n, int c)` that returns the solution of function $f(n) = n! + c$. Please do not use any helper function.
e.g. func(4,3) = 4! + 3 = 27

[8 marks]

ii. Why does a recursive function use the stack?

[1 mark]

iii. Explain how stack memory is managed when func(4, 3) is executed.

[4 marks]

[Total for Question 3: 18 marks]

Complexity Notation

Question 4

(a) What is the definition of $f(n)$ being in $O(g(n))$?

[1 mark]

(b) What is the definition of $f(n)$ being in $\Omega(g(n))$?

[1 mark]

(c) Please prove that $n^3 + 10n^2 + 10000$ is in $\Theta(n^3)$.

[4 marks]

(d) Please prove that $n^3 + 10n^2 + 10000$ is not in $O(n^2)$.

[1 mark]

(e) Given that $f(n) \in O(n^2)$ and $g(n) \in O(\log n)$, please prove that $f(n) * g(n) \in O(n^3)$.

[4 marks]

(f) f is a function that satisfies the following:

- f is in $O(n^2)$,
- f is in $\Omega(n)$,
- f is neither in $\Theta(n)$ nor in $\Theta(n^2)$.

Can you give an example of such a function f? Please also prove that the function you named indeed satisfies all of the above.

[5 marks]

[Total for Question 4: 16 marks]

Sorting and Searching

Question 5

(a) Please illustrate the process of sorting the list $\{5, 1, 6, 4, 9\}$ using bubble sort.

[2 marks]

(b) Please illustrate the process of merging the two sorted lists $\{1, 1, 5, 9\}$ and $\{4, 7, 12, 14\}$ in mergesort.

[2 marks]

(c) i. Given a list of n integers, you are asked to sort them in **descending** order using *quicksort*. Please write down the pseudo-code of quicksort with the last element as pivot.

[5 marks]

ii. The performance of quicksort depends on the selection of the pivot value. What kind of pivot value will result in the worst-case performance? Please provide some analysis.

[2 marks]

(d) Given a list of n values of type `int` (sorted, in **descending** order), please provide the pseudo code of binary search to find out whether the value `obj` is in the list.

[4 marks]

(e) Consider the following sorting algorithm (called "TwoMinSort"):

Let L be a list of distinct integers.

- Scan L to find the minimal value min and the second minimal value min'
- Swap the positions of min and $L[0]$
- Swap the positions of min' and $L[1]$
- Run "TwoMinSort" recursively on elements from $L[2]$ to $L[n]$

Please analyze the above algorithm and state its time complexity in Big-O notation.

[5 marks]

(f) Consider the following modified version of binary search:

Let L be a list of sorted values and let n be the number of elements in L:

- Check $L[n/3]$
- The above values determine which sublist to focus on (it should be noted that one sublist has size $n/3$ while the other sublist has size $2n/3$)
- Run the same algorithm recursively on the sublist

What is the time complexity of the above algorithm? Please support your answer with a brief proof.

[5 marks]

[Total for Question 5: 25 marks]

Linked Lists

Question 6

Define a linked list containing n nodes as follows:

```
struct Node {
  int data;
  Node *link;
}
```

(a) What is the time complexity for adding a node at the end of the linked list? Please also provide the pseudo-code for this operation.

[4 marks]

(b) Stacks and Queues are often implemented based on linked lists.

i. What is a stack?

[1 mark]

ii. What are the common operations of the queue?

[2 marks]

iii. What does FIFO represent in the context of algorithms and data structures?

[1 mark]

(c) Given a doubly linked list, what is the time complexity for deleting a node in the middle of the linked list?

[2 marks]

(d) Please describe how to swap two adjacent elements by adjusting only the links (and not the data) using:

i. Singly linked lists

[2 marks]

ii. doubly linked lists

[2 marks]

(e) A deque is a data structure consisting of a list of items, on which the following operations are possible:

- `push(x)`: Insert item x on the front end of the deque.
- `pop()`: Remove the front item from the deque and return it.
- `inject(x)`: Insert item x on the rear end of the deque.
- `eject()`: Remove the rear item from the deque and return it.

How do you use the singly linked list to implement a deque which support the basic operations above to be done with $O(1)$ complexity? Please provide C++ code segments and analysis to support your design.

[8 marks]

[Total for Question 6: 22 marks]

Trees

Question 7

Define a `tree` node as follows:

```
struct Node {
  int data;
  Node *left;
  Node *right;
}
```

(a) What is a tree in the context of algorithms and data structures?

[1 mark]

(b) What is the definition of a binary search tree?

[3 marks]

(c) Write a function `bool search(struct Node *root, int obj)` that takes as input a binary search tree root and a value of `obj`. The function returns whether `obj` exists in the tree or not.

[3 marks]

[Total for Question 7: 7 marks]

End of exam