Solution to Primary Examination, Semester 1, 2022

Seth

June 10, 2024

Inheritance and Object-Oriented Programming

Question 1

(a) Explain how the function getMin() in the class below demonstrates encapsulation/information hiding.

```
class myClass{
    // list of numbers stored as strings
    std::vector<int> myNumbers;

    public:
        int getMin(); //returns the smallest number in the list
}
```

(b) Explain the difference between a virtual and pure virtual function (i.e. virtual myFunction() vs virtual myFunction() = 0). In your

answer include an explanation of when each shoudl be used and the effect on child classes.

(c) The code below prints nothing when we want it to print "Roar". Explain what is wrong and how to correct the code.

```
class Animal {
    public:
        virtual void makeSound(){
    std::cout << "";
    }
};
class Tiger : publicAnimal {
    public:
        void makeSound() {
            std::cout << "Roar";
        }
};

int main (void) {
    Animal zoo[10];
    int zooSize=0;
    Tiger tigger;
    zoo[zooSize] = tigger;
    zooSize++;
    for (int i=0; i<zooSize;i++){
        zoo[i].makeSound();
    }
}
```

Recursion

Question 2

(a) Even if a recursive function has a base condition, it may not successfully terminate. Explain how this could happen?

(b) Write a head recursive **C++** function to compute the factorial of n: $n! = n \cdot (n-1) \cdot (n-2) \cdot \ldots \cdot 3 \cdot 2 \cdot 1$

(c) Write a head recursive **C++** function to compute the cumulative sum of n: $\text{sum} = n + (n-1) \cdot (n-2) + \ldots + 3 + 2 + 1 + 0$

Linked lists, Stacks, Queues

Question 3

Let class Node be defined as follows, and consider a linked list class that uses Node objects for storing the items. The linked list class has a member variable **Node* head**, that points to the first node of the list.

```
class Node{
    private:
        int data;
        Node* link;
}
```

(a) Linked list must have access to the data and link variables of the nodes in the list, explain what changes to the Node class

declaration should be made to allow a LinkedList class to acess these without making them public to other classes.

(b) Write a C++ push function for a Stack implemented as a linked list.

(c) Write C++ code for a member function of linked list class **void RemoveMin()**, that removes the node with the smallest value stored in the linked list.

Complexity

Question 4

(a) Indicates if the following statement is **true** or **false** and give a brief explanation to justify your choice.

"Ω *is the best case complexity of an algorithm.*"

(b) What is the time complexity of the following code segment? Briefly explain.

(c) Find a tight upper and lower bound for function $3n^2 - 5n + 7$, i.e., find $g(n)$ where $3n^2 - 5n + 7 \in \Theta\left(g(n)\right)$. Also, prove that the following statement holds : "$3n^2 - 5n + 7 \in \Theta\left(g(n)\right)$."

Sorting and Searching

Question 5

(a) You want to sort a list of names, each consisting of first and last names. People may have the same first name or last name, but the combination of both is unique.

This should be sorted by first names. For repeated first names, sort them by last name (e.g., John Blake, John Smith)

Assume that the list already sorted by last names. Answer the following questions.

i) Identify an appropriate sorting algorithm to complete the process, i.e. to sort by first name. Briefly justify your choice of sorting algorithm.

ii) Write the steps or high-level pseudo-code of the sorting algorithm you identifier in the previous sub-question.

iii) Identify a sorting algorithm that will **NOT** work, i.e., the list will not be sorted by first then last name. Briefly justify your answer and provide an example.

(b) Compare quick sort and merge sort in terms of time and space complexity (consider best/average/worst cases).

(c) Illustrate the process of sorting the list $\{1, 2, 3, 4, 5, 6, 7\}$ using quick sort to achieve the fastest runtime.

(d) You are working with a large array of data containing only single digits as values (i.e., $0, 1, 2 \ldots 8, 9$). The data is not ordered

in any way and may contain duplicates. You must be able to determine if a given value appears in the array or not.
You decide to use binary search to improve your worst case search complexity to $O(n \log(n))$.

i). How must you transform this data before you can use binary search?

ii). What is the most time-efficient method that we covered in this course for transforming the data? What is the complexity of this method?

iii). Illustrate the process of the binary search algorithm on the list $\{19, 15, 14, 10, 6, 5, 1\}$, start $= 0$, end $= 6$, and item $= 1$.

(e) Can the time complexity of sorting algorithm be lower than $O(n)$? Briefly justify your answer.

(f) Identify one non-recursive, comparison-based sorting algorithm where the best and worst cases have different complexity. Explain why they differ.

Trees

Question 6

Our primary examination (Semester 1, 2024) should contain sections on **Red-Black Trees** (even Trie) and Heap (heapify, min and max heaps, heapsort).

Solutions

1.

(a)