CS2040C Semester 1 2020/2021

Data Structures and Algorithms

Tutorial+Lab 01

Basic C++, Basic OOP, Analysis, Hands-on 1

For Week 03

Document is last modified on: August 21, 2020

1 Introduction and Objective

The purpose of this first tutorial+lab e-session is to recap the first four sessions of CS2040C: Introduction, basic C++ (esp basic OOP), basic analysis of algorithm, and to ensure that all students can code a simple C++ program using their own computer/laptop at home. The first half of the e-session is generally the 'tutorial' part and the second half of the e-session is generally the 'hands-on/lab' part. The tutors will control the timings and they don't have to divide the sessions exactly by half. There will be a short break during the transition.

As this is the first e-session, we will do a quick ice breaking at the start of the e-session. There are five+one different tutors this semester: B02-Wira Azmoon Ahmad, B03-Peh Yu Xiang, B07-Walter Kong, B08-Arjo Chakravarty, and B09-Ivan Chew Teck Meng with Nigel Neo Neng Kai as grader.

To get the most out of the tutorial part of these e-sessions, please try out all the questions in the tutorial component and give some answer even if you encounter difficulties in answering some of them. Before, during, or after the tutorial session, don't hesitate to clear up all doubts and questions you might have, with the tutor.

Every week, you will try to solve one selected Kattis problem during the 'hands-on/lab' component. The tutors already know the selected Kattis problems for this semester. However, these selected problems will be revealed to you on the spot each week (if you happen to already solve it, then you are free to just leave the e-session or actually you can stay back to help your peers). Tutor will guide all students to get (near) Accepted solution for each problem. These problems are not graded but attempting them during the hands-on time (and possibly to fully complete them afterwards) is beneficial to better understand CS2040C material.

There is no lab or tutorial participation marks this time due to the sessions being fully online.

2 Tutorial 01 Questions

C++ OOP (basic)

Q1). You are given a simple C++ program below:

```
#include <iostream> // you can change 'iostream' with 'that one'...
using namespace std;

class ListArray {
private:
  int N;
  int A[10]; // question 1a
public:
  ListArray() : N(0) {} // question 1b
  int get(int i) {
    return A[i]; // question 1c
  }
  int search(int v) {
    for (int i = 0; i < N; ++i)
      if (A[i] == v)
        return i;
    return -1;
  }
  void insert(int i, int v) {
    if ((N == 10) || (i < 0) || (i > N)) return; // question 1d
    for (int j = i; j <= N-1; ++j) // question 1e
      A[j+1] = A[j];
    A[i] = v;
    ++N;
  }
  void remove(int i) {
    for (int j = i; j < N-1; ++j) // question 1f
      A[j] = A[j+1];
    --N;
  }
  void printList() {
    for (int i = 0; i < N; ++i) {
      if (i) cout << " ";
      cout << A[i];
    }
    cout << endl;
  }
  void sortList() { // sort array A, question 1g





  }
};

int main() {
  ListArray* LA = new ListArray(); // question 1h
  LA->insert(0, 5);
  LA->insert(0, 1);
  LA->insert(0, 4);
  LA->insert(0, 7);
  LA->insert(0, 2); // we should have A = {2, 7, 4, 1, 5} by now
  cout << LA->get(3) << endl; // 1, A[3] = 1
  cout << LA->search(4) << endl; // 2, A[2] = 4
  cout << LA->search(6) << endl; // not found, -1
  LA->remove(1); // we should have A = {2, 4, 1, 5} by now
  cout << LA->search(4) << endl; // 1, A[1] = 4 now
  cout << LA->search(7) << endl; // not found, -1
  LA->printList(); // unsorted
  LA->sortList(); // we should have A = {1, 2, 4, 5} by now
  LA->printList(); // sorted
  return 0;
} // please copy paste the code above, test compile, and run it yourself
```

This code will be revisited soon during discussion of List ADT (read `https://visualgo.net/en/list?slide=2-1` until 2-8). For now, please answer the following sub-questions (please refer to the comments in the code above):

(a) Anything wrong with this line?

(b) What this line means?

(c) Any potential issue with this line? (also see question 1d below)

(d) What this line means?

(e) Any potential issue with this line?

(f) Any potential issue with this line?

(g) Implement this routine using any sorting algorithm that you know!

(h) Can we just write `ListArray LA;` in this line?

Analysis/Order of Growth

Q2). What is the bound of the following function? $\mathbf{F}(n) = \log(\mathbf{2}^n) + \sqrt{n} + \mathbf{100\,000\,000}$

1. O(n)
2. O($n \log n$)
3. O(n^2)
4. O(1)
5. O(2^n)

Q3.a). What is the bound of the following function? $\mathbf{F}(n) = n + \frac{1}{2}n + \frac{1}{3}n + \frac{1}{4}n + \ldots + \mathbf{1}$

1. O(2^n)
2. O(n^2)
3. O($n \log n$)
4. O(n)
5. O($\log^2 n$)
6. O($\log n$)

Q3.b). What about $\mathbf{G}(n) = n + \frac{1}{2}n + \frac{1}{4}n + \frac{1}{8}n + \ldots + \mathbf{1}$

3 Note

Remember that outside the official tutorial+lab hours, all tutors will stand by at their designated consultation time slot + e-venue (Zoom session) each week (unless they mention some exceptions) to answer CS2040C related queries. This information can be found at `https://www.comp.nus.edu.sg/~stevenha/cs2040c.html`, scroll to 'registration' section. Remember that you do NOT have to be in that tutor's class to join a tutor's 2h-consultation slot. It is a free-and-easy unstructured session. Anyone who wants to study together with the tutor or have any CS2040C related question(s) can join.