CS2040C Semester 2 2018/2019
Data Structures and Algorithms

Tutorial 01 - Basic C++, Basic OOP, Analysis
For Week 02

Document is last modified on: January 21, 2019

1 Introduction and Objective

The purpose of this first tutorial is to recap the first three sessions of CS2040C: Introduction, basic C++ (esp basic OOP), basic analysis of algorithm. For tutorial group 08 which – for some unknown reason – is the only tutorial scheduled BEFORE Tuesday lecture, parts about basic OOP and analysis of algorithm will still be alien to you.

Btw, as the tutorial group that you attended may not be your actual tutorial group yet, tutor will only briefly introduce him/herself and we will do proper introduction on Tutorial 02 on Week 03.

To get the most out of the tutorial sessions, please try out all the questions in the tutorial 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.

Btw, the tutorial participation marks to be given by the tutor **at the end of the semester** are simply as follows:

- 0% if you only attend ≤ 5 out of 10-11 tutorial sessions (CNY/Good Friday will affect all/some CS2040C tutorials this semester),
- 1% for **at most the bottom three** most-passive students (assuming these students attend > 5 tutorial sessions),
- 3% for **at least the top three** most-active students (answering questions when asked by TA – the correctness of your answers are secondary; or even just by asking your own questions to TA before/during/after class); in each tutorial group, and
- 2% for the rest.

2 Tutorial 01 Questions

C++ OOP (basic)

Q1). You are given a simple C++ program that will be revisited soon during discussion of List ADT:

```
#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
```

Now 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?

Another C++ Features Introduction

Q2). Some students of the class can suggest a few problems of `https://nus.kattis.com` that have been mentioned since Week -02 (01 Jan 2019) - the day of your tutorial to be discussed. The top ONE problem that still bother most of the students in class will be discussed (again). Tutor will use this section to gauge the situation at the ground and report to the lecturer for live coding speed adjustment, etc.

Analysis/Order of Growth

Q3). 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 hours, the lecturer and all tutorial TAs will stand by at their designated time slot + venue 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.