Monday, 28 January 2019

LAB DEMO 02

Lab TA: Sidhant Bansal

Lab TA Introduction & Expectations

My technical background

- 2x times Lab TA for CS2040C
- NUS ICPC Team Member
- Taken courses CS3233 + CS3230

Contact me at

- Facebook (profile pic —>)
- sidhant.bansal@u.nus.edu.sg

My expectations:

- Perfect attendance for all Lab Demos
- Each of you contribute something in those sessions
- The 3% participation points are somewhat subjective!*

C++ Compiler used by Kattis (2019)

Kattis uses C++17 standard

https://nus.kattis.com/help/cpp

- You can use #include <bits/stdc++.h>, albeit not SE industry standard
- You can use auto (range based loop)
- You can use lambda expression (e.g. as comparison function for sorting)
- You can use this kind of initialization: vector<int> A = {1,2,3};
- No more gets (deprecated since C++14)
- Structured bindings, better than C++11, e.g. for tuples (autos and ties)
- Steven (in lectures) and Lab TA/myself (in Lab Demos) will show lots of demonstration cpp code this sem, most in C++11 standard, but a few will be in C++14/C++17 (we will give remarks as not all compilers support the newest standards yet)

The Problem Sets

Steven's CS2040/C PSes (PS1-5) have this system

- Task A is the easiest/maybe not PE level, some/large points
  - Almost everyone are expected to solve this
  - Algorithm mentioned in tutorial/lab demos
- Task B is the medium/PE level, some points
  - Majority are expected to solve this
  - Algorithm mentioned in tutorial/lab demos
- The C is usually quite challenging, but low point(s)
  - Not all are expected to solve this
  - PE probably not going to be this hard
  - Knowing when to give up is important… otherwise your OTHER modules/aspects of your life will be in trouble

C++ STL vector

- ***constructor (init our vector with ‘reasonable’ size upfront)***
- ***[] operator (no need to use at)***
- *push_back*, *pop_back*
- *insert*, *erase*
- *front*, *back*
- ***begin, end***
- *assign*, *empty*, reserve, resize
- http://en.cppreference.com/w/cpp/container/vector

C++ STL algorithm

- **sort**, partial_sort, **stable_sort**
- *reverse*
- unique
- nth_element
- lower_bound, upper_bound
- **swap**
- random_shuffle
- **min, max**
- min_element, max_element
- http://en.cppreference.com/w/cpp/algorithm

C++/Java OOP, Revisited

- From Tut01
  - **Quick** review of the working solution

C++/Java OOP, Revisited

- From Tut01
  - Now changing the underlying data structure :O
    - https://visualgo.net/en/list?slide=2-4 with Java ArrayList (mentioned in https://visualgo.net/en/list?slide=2-7)
  - To illustrate the concept that ADT *may* be implemented with more than one data structure

Working C++ Code for VisuAlgo Ex

- https://visualgo.net/en/sorting?slide=1-2
- Pick one of application 1-6 and Lab TA will **code** the solution on the spot for you ☺️

Birthday Reminder Problem

- CP3.18b, page 16, task 5: Given the distinct and valid birthdates of n people as triples (DD, MM, YYYY), order them first by ascending birth **months** (MM), then by ascending birth **dates** (DD), and finally by ascending **age**.

VisuAlgo Online Quiz Training Mode

Make sure that you understand the explanation in:
https://visualgo.net/en/sorting

You can use VisuAlgo Online Quiz training mode to check your basic understanding about Sorting on "infinite"* number of random questions:
https://visualgo.net/training?diff=Medium&n=5&tl=5&module=sorting

PS1 Status (as of today :O)

https://nus.kattis.com/sessions/tm8b74/standings

- Public standings
- Lab TA will verbally say the status of full standings
  - Secret view, will not be shown
- Lab TA will give high level overview of the 3 tasks

Practice is important for these 5 PSes (15%) + end of semester PE (12%)

Don’t hesitate to contact me for more help if you need it