

CS2040C Lab Demos

Lab TA: Sidhant Bansal

Friday, 26 January 2018

(Need to start on Week 02 as we will lost 2 other Fridays: Week 05/CNY and Week 10/Good Friday)

LAB DEMO 01

Lab TA Introduction & Expectations

My technical background

- NUS ACM ICPC Regional Participant(2017)
- [IOI Bronze Medallist](#) (2017)

Hobbies - Chess, Competitive Programming, Badminton

CS2040C Lab TA



Incase of queries contact me at -

- sidhant.bansal@u.nus.edu
- Fb

My expectations:

- Perfect attendance for all $11-2+1 = 10$ Lab Demos
 - Lab 01 moves to Week 02; No Lab on Week 05+10 (we have 10 sessions only :o)
- Each of you contribute something in those 10 sessions
 - Answering my questions, presenting solutions, etc...
- The 3% participation points are somewhat subjective!*


Ice Breaking

When your name is called: stand up and say one sentence about yourself that is *very unique* about you, e.g - I am extremely bad at singing.

- Today, I will memorize at least 4 names
- Next meeting, I will memorize at least $4+4 = 8$ names
- Hopefully I will be able to remember everyone soon

Mini Game System

Quick Discussion of CS2040C Game System Features

- URL: <http://www.comp.nus.edu.sg/~stevenha/cs2040c.html#roster>
- We use this small-scale **gamification** in this module
 - The problem statements for PSeS are directly set-up in Mooshak
 - Solve those problems by submitting C++ code to Mooshak
 - This roster section is to view the list of achievements that you have obtained throughout this semester, for personal pride only

- Only for student with at least 1 achievement

Mooshak Online Judge System

We need another system for automatic grading

- Important URL: <http://algorithmics.comp.nus.edu.sg/~mooshak>
- Instant grading!
 - Typical Online judge verdicts: AC(cepted), W(rong)A(nswer), T(ime)L(imit)E(xceeded), R(un)T(ime)E(rror), C(ompile)T(ime)E(rror), Invalid Function, Program Size Exceeded (NEW)
- Unless there are special cases, if you get your code AC (Accepted), you will get that amount of points as stated in the problem description
 - However, post-deadline penalty (e.g. your code are found to be a very similar copy of someone else's code) can still alter the score
 - ${}_{41}C_2$ pairwise comparison check is a “small” number

C++ Compiler used by Mooshak

We use C++11 standard

- You can use `#include <bits/stdc++.h>` (beware - Not apt from the perspective of software developer)
- 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 guarantee on C++14/17 stuffs, I think it won't compile)
- Steven (in lectures) and myself (in Lab Demos) will show lots of demonstration cpp code this sem

The Problem Sets

Steven's CS2040C PSet (PS1-5) have subtask system

- Subtask A is always the easiest, but low -- non zero -- points
 - Everyone are expected to solve this
 - Algorithm mentioned in tutorial/lab demos (usually in tutorial)
- Subtask B (or also C) is/are CS2040C standard, medium points
 - Majority are expected to solve this
 - Algorithm mentioned in tutorial/lab demos (usually in lab demos)
- The last Subtask is quite challenging, but low (or zero) point(s)
 - Minority are expected to solve this
 - No need to feel bad if you cannot solve this part, it is a teaser of what can be done at higher level, when you know more algorithms 😊

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++ STL vector

- **constructor**
- **at** or **[]** operator
- **push_back**, *pop_back*
- *insert*, *erase*
- *front*, *back*
- *begin*, *end*
- **assign**, *empty*, *reserve*, *resize*
- <http://en.cppreference.com/w/cpp/container/vector>

C++ string

- **constructor or = operator**
- **at or [] operator**
- + (*concatenation*)
- ==, < (*comparison*)
- *find*
- *substr*
- *c_str*
- http://en.cppreference.com/w/cpp/string/basic_string

C++ Code Review of Past Demos

| When | Kattis Title | Purpose |
|-------|--------------|--|
| -01 | hello | Simple O of I/O |
| -01 | judgingmoose | Selection statement |
| -01 | timeloop | Simple I/O, loops (C and C++ have the same loop styles, except maybe 'auto' - to be shown later) |
| -01 | mia | Simple function |
| 1a | statistics | Array (min_element/max_element), or on the fly computation without array (min, max) |
| 1a | treasurehunt | 2D array; recursive function; (trying not to use global variable) |
| 1b | PS0 A+B | Concept of data types and their ranges |
| 1b | zamka | Concept of function, doubles as simple discussion of time complexity analysis |
| 1b | bookingaroom | Array of Boolean, or Vector of Boolean, <i>forced</i> introduction to C++ class (OOP) |
| 1b/2a | autori | C++ string, istringstream |
| 2a | PS0 C | More clever input parsing, or get the line and tokenize, be careful of terminating condition |
| 2a | register | Simulation, small array, to be used for another time complexity analysis |

Feedback

- Was I too slow / too fast ?
- Was it easy / hard ?
- Any other suggestions ?
- Should I demonstrate the code live or come prepared and go through it.