CSE100 Algorithm Design and Analysis
Lab Policies and Guidelines, Fall 2021
Last update: 08/27/2021

In each lab assignment, you will be asked to implement a specific algorithm that you have learned in class (in most cases). *You must follow the input/output formats* that are specified in each lab assignment description. We will provide an example in the description, but for more examples, please see t*, o* files under 'testfiles' directory you can find after upzipping labXX.zip. Each tx file contains an input, and the corresponding output is in file ox.

Submission Before the posted deadline, submit your source code through the assignments page of CatCourses. *You must submit only your cpp file (which includes the main function) and the file name must be (your ucm email id).cpp* — we apologize for this restriction, but we need to make our grading simple because of the large class size. You will have about 1-2 weeks to complete your assignment from the release date. If your submission is one week overdue, you will lose 10% of your points. You will get NO points if you submit one week after the deadline. You're allowed to resubmit as many times as you want.

Grading We will compile your code by standard c++ 14 compiler; so "g++ -std=c++14 ...". You can program your code in your favorite programming environment, but *make sure that you compile it by standard c++ 14 library and test it on a lab machine before submission*; we will give you a testing toolkit. Here's a trouble you could run into when you use a different compiler. Some students preferred to use Visual Studio, so programmed their code using it — well, it's a great tool, but you may not be able to use it after graduation unless your company buys it for you. Don't expect such luxuries particularly when working for start-ups. But the problem was that VS used a 'strengthened' compiler, which took care of some cumbersome issues such as initialization. So, even if the students compiled and ran their code successfully using VS, they couldn't pass any test examples. When we compiled their codes by standard c++ 14, their code output completely wrong answers due to a small initialization mistake.

We will run an *automatic* grader to test the correctness of your code. We will test it against 10 examples. Your score will be proportional to the number of test examples that your code passes. We will *read* your code to see if you actually implemented the algorithm you were asked to implement; if you implement a different algorithm, you will get zero points. Please help us by adding some *comments* to your code.

We will grade *twice*. First, right after the deadline, and then, one week after the deadline. Your scores will be updated on catcourses. Please wait 1-2 days for the update.

Your code should compile by standard c++ 14 and run correctly on any machine. If not, it means your code is not correct, or not robust enough, to say the least. *Your minimum goal should be to ensure that your code runs successfully on a UCM Linux machine (lab machine).* We may test your code on any machine of our choice for convenience. If you find your score lower than you expected, here's the protocol you should follow:

1. Come to your LAB to speak with your LAB TA (no exception, no virtual meeting).

2. Download the file you submitted from catcourses.

3. You must demonstrate that the downloaded file compiles using the standard c++ 14 compiler and pass the 5 test examples you're given.

   - If you used a different compiler, we won't do the next, meaning that you will get at most 50% for the lab.
   - Else, you must show what changes must be made to make your code run on a lab machine, and send it to the TA; the TA will test your file for all 10 examples
     - If no changes are needed, you will get the full score (possibly minus the overdue penalty)
     - Depending on the amount of changes needed, the TA reserves the right to take off 10-20% of the lab full score. *We expect that you test your code on a Lab machine before submission.*

How to Test your Code by Yourself We will provide you with the same automatic grading tool for you. The only difference is that we will disclose only 5 examples while we will test your code with 5 additional examples. By checking your code with the grader on a UCM Linux machine before you submit, you will be able to know if your output format is correct.

This is how you can use the grading tool, which is a shell script — so you need to use Linux, Mac terminal (if you're a mac user), or cygwin (If you're a window user). Or you can use a UCM Linux machine (in your lab or in a remote lab[1])

To test your code, unzip "labXX.zip" to your working directory and go inside the labXX folder. You will see several files. File "Grade" is the file you will use to test your code. Your execution file must be named "a.exe". Perhaps you may need to use chmod to change the permissions of "Grade". Towards this end, you can just type

"chmod 700 Grade" or "chmod +x Grade".

Now if you run "./Grade" and your implementation is correct, you will see the following messages:

```
Correct for 1 th example.
Correct for 2 th example.
Correct for 3 th example.
Correct for 4 th example.
Correct for 5 th example.
```

You can also find a summary of the execution in the file named "result". If you want to test your code for just one test file, you can try: ./a.exe < ./testfiles/t1

Finally, we also provide a very simple "Makefile" for you, which is very convenient for compiling your file(s). If you type "make," it will compile (your ucm email id).cpp; you should edit Makefile to rename yourid.cpp by (your ucm email id).cpp. If you type "make clean", then a.exe will be deleted. You don't have to use the provided "Makefile", but if you don't know how to use it, this is a good time to learn it. You can find abundant examples on the web.

[1] `https://it.ucmerced.edu/remotelab`

Academic Integrity For your convenience, we reproduce the academic integrity policy stated in the Syllabus here.

The campus Academic Honesty Policy states: "Academic integrity is the foundation of an academic community. Academic integrity applies to research as well as undergraduate and graduate coursework. Academic misconduct includes, but is not limited to cheating, fabrication, plagiarism, altering graded examinations for additional credit, having another person take an examination for you, or facilitating academic dishonesty or as further specified in this policy or other campus regulations.

Cheating is the unauthorized use of information in any academic exercise, or another attempt to obtain credit for work or a more positive academic evaluation of work through deception or dishonesty. Cheating includes, but is not limited to: copying from others during an examination; sharing answers for a take-home examination without permission; using notes without permission during an examination; using notes stored on an electronic device without permission during an examination; using an electronic device to obtain information during an exam without permission; taking an examination for another student; asking or allowing another person to take an examination for you; tampering with an examination after it has been corrected, then returning it for more credit than deserved; submitting substantial portions of the same academic work for credit in more than one course without consulting the second instructor; preparing answers or writing notes in a blue book before an examination; falsifying laboratory, or other research, data or using another person?s data without proper attribution; allowing others to do the research and writing of an assigned paper (for example, using a commercial term paper service or downloading a paper from the internet); and working with another person on a project that is specified as an individual project.

Plagiarism refers to the use of another's ideas or words without proper attribution or credit. This includes, but is not limited to: copying from the writings or works of others into one's academic assignment without attribution, or submitting such work as if it were one's own; using the views, opinions, or insights of another without acknowledgment; or paraphrasing the ideas of another without proper attribution. Credit must be given: for every direct quotation; when work is paraphrased or summarized, in whole or in part (even if only brief passages), in your own words; and for information which is not common knowledge. The requirement to give credit applies to published sources, information obtained from electronic searches, and unpublished sources.

Collusion is when any student knowingly or intentionally helps another student to perform any of the above acts of cheating or plagiarism. Students who collude are subject to discipline for academic dishonesty. No distinction is made between those who cheat or plagiarize and those who willingly facilitate cheating or plagiarism."
More information about the policy and the Office of Student Conduct can be found here:

http://studentconduct.ucmerced.edu/
Some students may still have some confusion (albeit the policy is quite clear), in particular concerning collaboration. The following rules are in place to make this issue clearer, from the perspective of my class.

Cheating vs. Collaboration.: Collaboration is a very good thing. On the other hand, cheating is considered a very serious offense. Please don't do it! Concern about cheating creates

an unpleasant environment for everyone. If you cheat, you risk losing your position as a student in the college. The school's policy on cheating is to report any cases to the university judicial office. What follows afterward is not fun. So how do you draw the line between collaboration and cheating? Here's a reasonable set of ground rules. Failure to understand and follow these rules will constitute cheating and will be dealt with as per university guidelines. The Simpson?s Rule: This rule says that you are free to meet with a fellow student(s) and discuss assignments with them. Writing on a board or shared piece of paper is acceptable during the meeting; however, you should not take any written (electronic or otherwise) record away from the meeting. This applies when the assignment is supposed to be an individual effort or whenever two teams discuss common problems they are each encountering (inter-group collaboration). After the meeting, engage in a half-hour of mind-numbing activity (like watching an episode of the Simpsons), before starting to work on the assignment. This will assure that you can reconstruct what you learned from the meeting, by yourself, using your brain. The Freedom of Information Rule: To assure that all collaboration is on the level, you must always write the name(s) of your collaborators on your assignment in the beginning of your submission file as a comment.

Computer Science Department Policy on Academic Honesty. As stated in the campus-wide Academic Honesty Policy (AHP), "academic integrity is the foundation of an academic community". Accordingly, the CSE faculty takes this matter very seriously and has embraced a zero tolerance on this matter. The process described in the following establishes the minimum consequences for violations of the AHP in CSE courses, but repercussions may be more severe for egregious violations. The Computer Science Department Policy on Academic Honesty ("CSE Policy" from now onwards), does not substitute the AHP but rather specifies how it will be implemented when students enrolled in classes offered by the Computer Science and Engineering (CSE) department are found in violation of the AHP. In particular, the CSE Policy defines how the CSE faculty implements the "Instructor-Led Process" described in AHP 802.00.A. This policy and the associated processes have been developed in collaboration with the Office of Student Conduct and the School of Engineering and is jointly implemented by the CSE Faculty, the School of Engineering, and the Office of Student Conduct. The CSE Policy has been in effect since the Fall 2019 term.

Preamble. Computer science education relies on a variety of methods to assess students' preparation and learning. The term "assignment" shall be interpreted as any method or process resulting in a grade or contributing to the final grade for a class. Accordingly, the term "assignment" used in the following includes, but is not limited to: homeworks, quizzes, in-class exams, take-home exams, programming assignments, software projects, and presentations.

Shared Responsibility. Maintaining an environment where academic integrity is valued and enforced requires commitments by both instructors and students. Instructors will specify what type of collaboration is allowed or disallowed for a given assignment, and students should strictly follow the provided guidelines. When in doubt, students should contact the instructor and ask for clarifications.

First Infraction. If it is determined that a student has cheated, plagiarized, or otherwise violated the AHP, the student will receive a 0 (or equivalent grade) for the assignment. As per the AHP, violations will be reported to the Dean of the School of Engineering and the Office of Student Conduct for review of possible violations of the Code of Student Conduct.

Additional Infractions. The School of Engineering keeps a record of all infractions reported by its faculty. If upon receiving a notification it is determined that the student has one or more prior violations of the AHP, the School will inform the instructor who reported the new violation. The additional violation will immediately lead to a failing grade (F) for the course. The student will be informed in writing and will not be allowed to withdraw from the class. According to CSE Policy, students should note that even the first infraction in a class may lead to a failing grade if after reporting it is determined that the student had been previously sanctioned for one or more infractions in other classes. Students will have the right to appeal the instructor's decision as per AHP 802.00.A. Resources If in doubt, students are encouraged to seek guidance from the faculty, advisors, and the Office of Student Conduct. Additional resources can be found on: `http://studentconduct.ucmerced.edu/` `https://ombuds.ucmerced.edu/` `https://eecs.ucmerced.edu/sites/eecs.ucmerced.edu/files/page/documents/computer_science_department_policy_on_academic_honesty_fall_2019.pdf`

Some examples of academic violations/non-violations.

1. **Searching an implementation of the algorithm you're asked to implement.** This is a violation. Of course, there's no way that we can know for certain if you quickly eyeballed other implementations. If you just did, your code will substantially different from what you saw. But the longer you stare at other codes, the more similar your code will be to those. Your risk will get elevated.

2. **Sharing your code with your classmates**. This is a violation. Even if you receive the code only for reference, it will considered as a violation.

3. **Do not post your implementations in public sites (e.g. github) and make them accessible to others.** Trust me. Implementing easy common algorithms won't impress anyone. And your classmates could be tempted to copy them. In particular, if you post your codes before the hard deadline, it will be a violation.

4. **Searching examples of common data structures (different from what you're asked to implement).** This is not a violation. For example, you might have forgot how to use linked lists. In this course, you may need them to implement some graph algorithms.

We highly recommend that you state any references you used in the beginning of your cpp file as comments.