

I been searchin', uh huh searchin', searchin' every which a-way

Programming project 2 for CISC 320 Algorithms

April 28, 2015

This project is an exercise in application of all the algorithm design techniques you have studied, along with the potential to choose, use, modify any specific algorithms and data structures we have studied, to achieve the goal. In a nutshell, the project is to solve one of the problems from the ACM Midatlantic regional programming contest of 2014 (see the handout or the link on our webpage). Any ONE of the five problems C, E, F, G, H is to be solved (A,B,D are excluded).

Now choice is good: It surely makes your life easier/better, right? ...To select your personal favorite, for whatever reason you may have, must be better than having no choice. But, have caution, when the going gets tough you can get caught up in waffling between the choices. I strongly recommend proceeding this way:

1. Read all five problems carefully and map out a high level sketch of your solution ideas for each one. OK, this is full bore waffling amongst the choices.
2. Take your first choice and write a careful, detailed description (pseudo-code) of your solution algorithm. A little waffling could be acceptable here if you realize your pseudocode has serious flaws.
3. Implement your solution. No waffling! No going back to stage one now. OK, you might find the need to revisit stage 2 and improve your pseudocode. Key Do and Don't: **Do** always have a detailed, written pseudocode showing the logic of your solution. **Don't**, at this coding stage, throw it all away and start over with another problem.

This project is to be written in C or C++.

You may work in teams of two or three. Submit your solution on Sakai. The solution must contain team member names. The members of a team must each submit, even though their submissions are identical. Send an email if your team is different than in project 1 (including if you are going solo where you were in a team for proj 1).

Deliverables:

- test suite Pretend you are judging the contest. Specifically, design test cases for your chosen problem with the idea to catch bugs that might be in anyone's solution regardless of solution details (blackbox testing). Consider corner cases such as instances that are as minimal as possible or as maximal as possible with regard to problem specification parameters. Of course, these tests will be useful to you in your own debugging. Due Wednesday, **May 6** on Sakai.
- pseudocode Present a detailed description of the solution algorithm for your chosen problem. This will be the basis for your coding efforts, so make it good. Due Friday, **May 8** on Sakai.
- code The final submission: code file(s) together with a Readme file telling how to build and run it. Due Friday, **May 15** on Sakai.

The first two deliverables are "for your own good," to get started in a timely way with some key tools of the job. They will be graded on a pass/fail basis. That is, while you may get a little feedback, your instructors will not be giving detailed critiques of your test suite or algorithm plan at this point. Turning in nothing on time or a meagre, "below threshold" submission for test suite and pseudocode will cost you. However slackers may shoot for the minimal effort and still get the pass. But don't do that. Like I said, for your own good, make a robust test suite and a thorough algorithmic plan by these deadlines.

Regarding the final submission: make it awesome and on time.