CS 520
Final project description

Final projects will be completed in teams of 4 or 5 students. Each team is responsible for a single project.
You should select a team and a project by **Tuesday, March 4, 2021, 9:00PM EST**.
Your mid-point report will be due **Tuesday, April 6, 2021, 9:00AM EDT**.
The final project will be due **Tuesday, May 4, 2021, 11:55 PM EDT**.
There are four options for a final project[1] (each team will do one):

1. MSR Mining Challenge
2. Replication Study
3. ML (Machine Learning) Development Toolkits
4. EleNa: Elevation-based Navigation

The first section provides more details about each of the four options for the project and the second section describes what needs to be included in the mid-point report for that project.

Topic selection

MSR Mining Challenge The Mining Software Repositories conference runs an annual challenge in which they provide a dataset and ask you to answer research questions about the dataset. Read the description of the past and current years' datasets, research questions, and challenges here:
https://2021.msrconf.org/track/msr-2021-mining-challenge#Call-for-Mining-Challenge-Papers
https://2020.msrconf.org/track/msr-2020-mining-challenge#Call-for-Papers

Replication Study A replication study takes an existing research paper, replicates its experiments on the same data, and then extends the experiments to expanding that data set on which the experiments are run. For this project, we highly recommend selecting a paper with publicly available dataset and code to execute the experiments. The project involves a write up describing the process of replicating the experiments, deviations in the achieved results from the original ones reported in the paper, and lessons learned from applying the experiments to new data.

Here is a list of several papers that are good candidates for replication:

1. Automatic generation of oracles for exceptional behaviors from Javadoc comments.
   Paper: https://dl.acm.org/citation.cfm?id=2931061
   Source code: https://github.com/albertogoffi/toradocu
2. EvoSuite: Automated test generation
   Paper: https://dl.acm.org/citation.cfm?id=2685612
   Source code: http://www.evosuite.org/ and https://github.com/EvoSuite/evosuite
   Dataset: https://github.com/rjust/defects4j

[1] In unusual cases, it is possible to convince the professor to do a self-defined project.

3. SimFix: Automated program repair

   Paper: https://xgdsmileboy.github.io/files/paper/simfix-issta18.pdf

   Source code: https://github.com/xgdsmileboy/SimFix

   Dataset: https://github.com/rjust/defects4j

ML Development Toolkits This project will involve applying an ML development toolkit to one or more data sets to produce the desired ML models. Additionally, the project will involve identifying connections between such ML development toolkits and Software Engineering topics covered in this course. The final deliverables are a presentation (and write up) of the lessons learned and experimental evaluation as well as any development artifacts (e.g., documentation, version control repository).

Alternatively, this project could involve comparing/contrasting two different ML development toolkits on the same data set.

Here are some ML development toolkits. There are many others available.

1. Neptune.ai: https://neptune.ai
2. Weights & Biases: https://wandb.ai/site
3. AI Fairness 360

   Paper: https://arxiv.org/abs/1810.01943

   Toolkit: https://aif360.mybluemix.net
4. Fairkit Learn

   Paper: https://arxiv.org/abs/2012.09951

   Toolkit: https://pypi.org/project/fairkit-learn/

EleNa: Elevation-based Navigation Navigation systems optimize for the shortest or fastest route. However, they do not consider elevation gain. Let's say you are hiking or biking from one location to another. You may want to literally go the extra mile if that saves you a couple thousand feet in elevation gain. Likewise, you may want to maximize elevation gain if you are looking for an intense yet time-constrained workout.

The high-level goal of this project is to develop a software system that determines, given a start and an end location, a route that maximizes or minimizes elevation gain, while limiting the total distance between the two locations to x% of the shortest path.

Components:

Your software system will most likely have four main components:

1. Data model that represents the geodata.
2. A component that populates the data model, querying, e.g., OpenStreetMap.
3. The actual routing algorithm that performs the multi-objective optimization.
4. A component that outputs or renders the computed route.

While all components are necessary for a working prototype, you may choose to focus on some of them in greater detail. For example:

- If you focus on developing and experimenting with several routing algorithms, it is sufficient to have a simple interface for entering the start and end location and a simple output that represents the route.

- If you focus on a sophisticated UI with proper rendering of the computed route, it is sufficient to have a basic data model and routing algorithm.

Resources:

- The A$\star$ algorithm: https://en.wikipedia.org/wiki/A*_search_algorithm
- Dijkstra's algorithm: https://en.wikipedia.org/wiki/Dijkstra%27s_algorithm
- OpenStreetMap wiki: http://wiki.openstreetmap.org/wiki/Main_Page
- The following paper, in particular Section 2, provides a very accessible introduction and overview of metaheuristic search algorithms:
  https://pdfs.semanticscholar.org/9c83/752460cd1024985981d4acaa7bc85e15c0f7.pdf

Mid-Point Presentation

On Tuesday, April 6, 2021, 10:00AM EDT, during class time, you will do a 7-minute presentation and describe your project. We will use the beginning of the next class if necessary. The presentation times will be chosen randomly to be fair to everyone.

The time limit is strict, and you will be expected to tell us four elements of your project:

1. **The Problem.** Tell us what you are going to build. If you are doing a research-focused project, tell us the research question(s) you will answer. If you are building a system, show us a prototype or describe the basic functionality and where your work will focus. Keep it focused. One slide. No more than 2 minutes total. Practice what you will say. Put a one-sentence summary on the slide.
2. **The Design.** Tell us how you will build what you are building. If you are building a system, tell us what technology you will use and show us the high–level architecture. If you are doing a research-focused project, tell us the design of your experiment(s). Again, no more than 2 minutes total. Practice what you will say. They key is to convince the audience you will succeed.
3. **The Evaluation.** Tell us how you will know you succeeded. If you are doing a research-focused project, tell us what data you will use, how you will know that your results make sense, what statistical tests you'll apply, etc. If you are building a system, tell us your testing plan and how you will execute it. Again, no more than 2 minutes.
4. **The Plan.** Tell us (really quickly) what your planned timeline is and each group member's responsibilities. Do not go into details; just show that you have a plan.

On Moodle, you only need to submit your presentation slides (as either PDF or PowerPoint).