Project Description

The project aims to implement algorithms to solve real-world problems and perform empirical evaluations. Try finding heuristics that speed up things in practice and find complex inputs to break the heuristics. See if you can prove that the heuristics work well for some critical classes of input. Try to find modifications of how the algorithm was initially described to obtain speedup and improved memory consumption (e.g., by constant factors). A typical implementation says, "We implemented algorithm X, and it was awful; then we added heuristic Y, which was great!".

Be cautious - implementations can take much time. Make sure what you are trying to implement is a reasonably "small" algorithm. Give serious thought to test inputs-ideally, they will come from real-world problems or be specially designed to "stress" some aspect of the algorithm. You probably also need a control (i.e., dumb) algorithm to compare yours to. You are encouraged to relate the final project to your research interests and will not be limited to the topics discussed in class.

What Topic Should I Work On?

The best topic to pick is one you are interested in anyway. Many of you are already involved in some research project; some thoughts may reveal an algorithmic component. Perhaps your system is presently using a naive algorithm for X and could be improved by using a more sophisticated one.

You can do background reading on the topic, develop a theoretical problem model, devise a solution, or take some extant algorithm and try it out. If you aren't working on anything, try browsing through what we've covered in class and papers (see below) to identify an interesting area. You are not limited to the topics we have covered in class. Anything that uses algorithmic ideas and a theoretical framework to devise more efficient solutions is fair game.

Here are some problems that are worth studying:

1. Graph coloring problem using exact/heuristic algorithms.
    a. Introduction to Graph Coloring
    b. Minimum Graph Coloring Problem
    c. Graph Coloring
    d. Graph Coloring Instances
    e. OR-Library
    f. Evolutionary Graph Coloring
    g. A Hybrid Approach for Exact Coloring of Massive Graphs
2. Shortest path problem and speed-up techniques
    a. Speed-Up Techniques for Shortest-Path Computations
    b. Fast Path
    c. Advanced-Shortest-Paths-Algorithms

d. Arc-Flags for Public Transit Routing

e. Shortest Path Distance Approximation using Deep learning Techniques

3. Knapsack problem using exact/heuristic algorithms

a. The Knapsack Problem

b. Genetic Algorithm – Zero One Knapsack

c. Knapsack Problem via Genetic Algorithm

d. Genetic Algorithm: — Knapsack Problem

e. Genetically solving the age old Knapsack Problem

f. Instances of 0/1 Knapsack Problem

g. Knapsack Problem Instances

4. Etc...

Where Can I Find Papers?

There are numerous sources of papers on algorithms. The best work in the area is published in three conferences, each with yearly proceedings:

- The ACM Symposium on Theory of Computing (STOC).
- The IEEE Symposium on Foundations of Computer Science (FOCS).
- The ACM-SIAM Symposium on Discrete Algorithms (SODA).

There are lots of other sources, of course. There are also journals, but they tend to be rather behind.

Policies:

- You may work in teams of up to two on the project.
- Team members might not receive the same grade on the project.
- You must submit a report (or a research paper) by the last day of week 15, which should be at most 10 pages, excluding the title page (which can include team member names, title, and abstract) and bibliography.
- Use at least 10pt font and 1-inch margins all around. A clearly marked appendix of unbounded length can follow the bibliography, but there is no guarantee that any of it will be read. As such, keep the most interesting parts of your write-up in the 10-page body.
- You are free to write your reports using Word or LaTeX. I encourage you to use LaTeX.
- You must upload your reports and source code via Canvas.
- If there are multiple members on the team, only one submission needs to be made (make sure all team members' names are written in the submission, though!)
- The final grade will depend upon the quality of the report, source code, and presentation.