CSCI 2500-C: Graph Theory and Algorithms
Meeting Time: Monday and Wednesday 3:00 - 4:20 (T hour)
CIT 101

Instructor: Philip Klein (`klein@brown.edu`)

Office Hours: By appointment.

Prerequisites: You must have passed a course on algorithms equivalent to CSCI1570 (*Design and Analysis of Algorithms*). An understanding of basic linear algebra is very helpful. If you have not studied linear algebra, you will have to do a bit of study on the side.

Readings: Research papers and book chapters, to be provided.

Assignments and work: Problems sets assigned roughly every two weeks. A midterm exam and a final exam. The midterm will take place in class on March 22. The final exam will be on May 11 at 9:00 am.

Grading: Your grade will depend on performance on homeworks and exams, and participation in class. You must perform at a passing level in all three categories in order to pass the course.

Time required: Time required for attending class, working on and writing up homeworks, and studying for and taking exams: roughly 14 hours per week.

Graphs (a.k.a. networks) are ubiquitous in computer science. Moreover, algorithmic problems on graphs have played a singular role in the development of theoretical computer science, e.g. the notions of polynomial time and of linear time and of NP-completeness. This course focuses on those aspects of graph theory that are most relevant to algorithms, on the classical algorithmic developments that have shaped the field, and on some emerging algorithmic methods that show promise of theoretical or practical impact.

Problem sets will be assigned roughly every two weeks. You are welcome to work on the homeworks with other students in the course. However, you must not retain any written (physical or digital) record from your collaboration, and must write up your solutions on your own, without assistance from others. In addition, before drawing on an online resource, you should check it with me. Most online resources are fine but those that provide ready-made solutions to assigned problems are not. Your homeworks will be graded on correctness *and on elegance and brevity of presentation.*

The course web page is `http://cs.brown.edu/courses/csci2500-c/`
That page in turn will link to the course calendar, which will indicate for each

lecture my best guess as to the topic (subject to change). The course page will also have links to the reading assignments. Homework assignments will be emailed to registered students.

Because the number of students will likely be small, there will be plenty of opportunity for discussion. I strongly encourage students to participate in such discussion. There will also be some opportunities for group problem-solving during lectures.

My intention is that the classroom be a welcoming environment for students. In the classroom and out, I expect my students to be kind and respectful.

The theme for this year is *obstructions*. Often the nonexistence of a graph-theoretical structure can be demonstrated by exhibiting a different graph-theoretical structure. For example, Hall's Theorem provides a simple reason that a given bipartite graph cannot contain a perfect matching: there is a subset of vertices whose size is greater than the number of the subset's neighbors. It is clear that a graph with this property cannot have a perfect matching. What is remarkable is that a graph without this property is guaranteed to have a perfect matching. Thus the existence or nonexistence of such a subset precisely predicts the nonexistence or existence of a perfect matching. Moreover, the attempt to find such a subset can lead one to develop an algorithm that is guaranteed to find a perfect matching if one exists.

We will focus on graph-theoretical properties that have corresponding obstructions. In most cases, there is a natural optimization (minimization or maximization) problem associated with the graph-theoretical property, in which case the obstruction also corresponds to an optimization (maximization or minimization) problem. We are interested in theorems that show an exact or approximate relationship between the value of the minimization problem and the value of the maximization problem.

To refer to this phenomenon, we borrow the term *duality* from linear-programming theory, and in this context we use the term *primal* to refer to the original optimization problem and use the term *dual* to refer to the second, related optimization problem. We will be especially interested in optimization problems for which consideration of the dual helps lead us to the design of an algorithm for finding an optimal or near-optimal solution to the primal.

On the following page is a rough list of topics from which we will select a *subset* to study. The items enclosed in square brackets are topics we will perhaps touch on but not explore in depth. I have outlined the early topics in greater detail.

- Bipartite graph matching
  - Hall's Theorem

  - Vertex Cover (König's Theorem)
  - Total unimodularity and integrality of polyhedral vertices
  - Weighted Vertex Cover (Egerváry's theorem)
  - Finding a maximum matching by augmentation
  - Hopcroft-Karp algorithm for nonbipartite nonweighted matching
  - Weighted matching (Hungarian method, Gabow-Tarjan scaling algorithm)

- Maximum flow and min cuts
  - Edmonds-Karp algorithm for maximum flow
  - [Use of link-cut trees in max flow]
  - [Subsequent work: Goldberg-Rao; using electrical flows]
  - [Total unimodularity]
  - [min-cost flow: Goldberg-Tarjan algorithm]

- Nonbipartite matching
  - Edmonds' algorithm for unit-weight matching
  - Edmonds' algorithm for weighted matching
  - Gabow-Tarjan scaling algorithm
  - Application to Postman problem
  - Christofides' approximation algorithm for traveling salesman problem
  - Application to shortest paths with negative lengths

- Cuts in graphs
  - Gomory-Hu cut trees
  - Algorithms for finding cuts and k-cuts: Karger-Stein, Karger, Thorup, and Kawarabayashi and Thorup
  - [Bisection and sparsest cut]
  - [Correlation Clustering]
  - [Metric labeling]

- Primal-dual method for approximation
  - Steiner forest
  - prize-collecting Steiner tree

- Surface-embedded graphs and minor exclusion
- Treewidth and branchwidth
- Graph embeddings
- [Erdős–Pósa Theorem]