MAS477 / CS492B, 2010 Fall, KAIST

Introduction to Graph Theory / 그래프이론개론

This course is an introduction to some of the major topics of graph theory. They include graph connectivity, matchings, planar graphs, graph coloring, and nowhere-zero flows.

Basic notions and theorems covered in Discrete Mathematics (MAS275 or CS204) will be assumed; but we will review them in the first week.

Lecture MWF 1PM-1:50PM Classroom: E6-1 (자연과학동), Room 2413

Instructor Sang-il Oum (엄상일) `http://mathsci.kaist.ac.kr/~sangil/`

Email: sangil@kaist.edu Office: E6-1 Room 3403.

Office Hours Tuesday 4PM or by appointments.

We will discuss homework solutions during the office hour following the due date. Therefore it is recommended to attend office hours.

Course website `http://moodle.kaist.ac.kr/`. (Passcode will be distributed in class)

Textbook Main textbook: R. Diestel, "Graph Theory", 3rd edition or 4th edition. Springer

`http://diestel-graph-theory.com/`

Reference: Bondy, Murty, "Graph Theory", Springer

`http://dx.doi.org/10.1007/978-1-84628-970-5`

Grading 20% Homework, 30% Midterms, 50% Final. (Bonus up to 15% for Programming Project)

The lowest score and the second lowest scores from assignments will be dropped. You will earn A if (but not only if) your score is at least 90, B if your score is at least 80, C if your score is at least 70.

Bonus points *up to* 15% will be given if one submits a computer program for some projects.

Midterm Exam To be decided. (Possibly on Oct. 30, Saturday afternoon.)

Final Exam Dec. 15, Wednesday 1PM-2:50PM (tentative).

There will be no make-up exams. Exams will be "closed book", "closed note". Calculators are not allowed in the exams. Any violation of honor code will be reported.

Homework Homework will be given weekly or biweekly in class on Wednesday. The assignment is due at the beginning of class on the following Monday. You may collaborate with other students. But **homework should be written by yourself independently and you must understand your solution.**

Plan **Week 1-2** Basics. Reviews. (chapter 1)

Week 2-4 Matchings (chapter 2)

Week 4-5 Connectivity (chapter 3)

Week 6-7 Planar graphs (chapter 4)

Week 8 Midterm Exam

Week 9-10 Coloring (chapter 5)

Week 11-12 Flows (chapter 6)

Week 13 Extremal Graph Theory (chapter 7)

Week 14 Ramsey Theory for Graphs (chapter 9)

Week 15 Graph minors and well-quasi-ordering (chapter 12)

Week 16 Final Exam

No lectures on : Sep 6, 8, 10 (conference trip), 22 (Chuseok), Oct 13 (business trip), Oct 18 (conference trip), Oct 20, 22, 25 (midterm exam period)

We will discuss the schedule to make up 5 lectures. (An idea: start a lecture 30 minutes earlier 10 times ; every Wednesday class)

- For week 13-15, we may cover alternative materials.
- Hint for the course: Definitions are very important!

  Attend the class, Ask questions, Do the homework, Solve exercise problems.

  You should learn how to prove mathematically. Most of the homework problems and exam problems will require you to prove something.

Instruction for optional programming project

1. Choose your favorite graph algorithms, discussed in class or not. (The bonus point will depend on the difficulty of the algorithm and completeness, clarity, and correctness of the implementation.)

   Examples:

   (a) Finding a maximum matching in a bipartite graph.
   (b) Finding a maximum matching in a graph.
   (c) Finding a maximum number of internally disjoint paths between two vertices.
   (d) Testing whether a graph is planar or finding a subdivision of K_5 or $K_{3,3}$.
   (e) Finding an ear decomposition of a 2-connected graph.

2. Write programs in python using networkx `http://networkx.lanl.gov/`

   If possible, it is recommended to make your program to additionally output a short certificate – the proof that explains why the result is correct.

   The program should have a separate "function" to perform the main algorithm. (Use "def" in python) For instance,

```
import networkx as nx
# main algorithm
def number_of_vertices (G):
    return len(G.nodes());

# input/output
G=nx.MultiGraph()
G.add_edge(1,2)
print number_of_vertices(G)
```

   (Please avoid mixing the input/output with the main algorithm part to make it reusable.)

3. Submit them to the professor. You should include the following.

   (a) Brief description
   (b) The source code of the program.

Other interesting courses in Fall 2010

- MAS581: Topics in Mathematics (Analytic Combinatorics) by Mihyun Kang (T.U. Berlin).

  Sep. 12 (Sunday)–17, Exam on Sep 19. `http://bit.ly/bV4giE`

- MAS478 Discrete Geometry by Prof. Andreas Holmsen. `http://mathsci.kaist.ac.kr/~andreash/mas478/`