COMP572–Combinatorial Optimization

- Instructor: Mordecai Golin (golin@cs.ust.hk)
- TAS:
  Zhou Zen (cszz@cs.ust.hk) & Leung Yiu Cho (cscho@cs.ust.hk)
- Meeting Times: Tuesday/Thursday 12-13:20
- Marking Scheme:
  3/4 Homework Assignments + a programming project
- Textbook:
  Combinatorial optimization : algorithms and complexity
  Christos H. Papadimitriou, Kenneth Steiglitz
  Englewood Cliffs, N.J. : Prentice Hall, c1982
- Reference Book:
  Introduction to algorithms (2nd ed)
  Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest , Clifford Stein
  Imprint Cambridge, Mass. : MIT Press ; Boston : McGraw-Hill, c2001

  and many others

Course Requirements

- Linear Algebra
  Vector spaces, bases, matrices, determinants
- A class in the
  Design and Analysis of Algorithms (COMP271)
  Graph Algorithms, Data Structure Analysis,
  $O()$ Notation, Dynamic programming,
  NP Conpleteness, etc..

  In particular, for new students:
  If you were told that you have to take COMP271 and have not been waived out of the requirement then you should **NOT** take this class.

What is this course?

- How to solve optimization problems *exactly*

  as opposed to heuristically, e.g, simulated annealing, genetic algorithms

- Algorithmically Oriented

  Some of this material is taught in optimization departments from a different perspective.

- The types of problems we will look at arise *everywhere*,

  e.g., database, networking, chip design, logistics, graphics, etc.

- This is a basic technique course.

  We will specifically **NOT** look at individual applications.

- This is an introductory PG course.

  It is **NOT** intended to teach cutting edge research techniques.

  It is designed to teach the old techniques upon which cutting-edge techniques are based.

Tentative Syllabus

- Network Flow (CLRS book)
- Binomial Heaps (CLRS book)
- Linear Programming and related topics (PS book)
  (50% of the class)
- Matchings (PS book)
- Matroids (PS book)
- Dynamic Programming Speedups (possible advanced topic)
- Introduction to Approximation Algorithms (possible advanced topic)

Two Typical Problems

Let $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ be a weighted graph
$\mathbf{V}$ are the vertices; $\mathbf{E}$ are the edges
For $\mathbf{e} \in \mathbf{E}$, $\mathbf{w}(\mathbf{E})$ is the weight of $\mathbf{e}$.
The weight of edge-set $\mathbf{E}' \subseteq \mathbf{E}$ is $\mathbf{w}(\mathbf{E}') = \sum_{\mathbf{e} \in \mathbf{E}'} \mathbf{w}(\mathbf{e})$.

A **tree** is an acyclic connected graph containing all V.
A **Hamiltonian Cycle** is a simple cycle containing all V.

- **Minimum Spanning Tree:**
  Find minimum-weight tree among all trees in $\mathbf{G}$.
  Polynomially time solvable ($|\mathbf{E}| \log |\mathbf{V}|$)
  e.g., Kruskal's Algorithm and Prim's algorithm.
- **The Travelling Salesman problem:**
  Find minimum-weight H.C. among all H.C. in $\mathbf{G}$
  NP-Hard, so polynomial-time solution is unlikely.