

About CS5234 (Wk 0)

- ✧ **Course Overview**
 - ❑ **All the Nitty Gritty**

- ✧ **About CS5234 Homeworks**
 - ❑ **Homeworks are fairly challenging**
 - ❑ **About Academic Honesty and Cheating**

- ✧ **Is this course suitable for you?**
 - ❑ **Course Pre-requisites**
 - ❑ **Mathematical Maturity**

CS5234: Comb and Graph Alg (Fall 2002)

✧ Instructor: Leong Hon Wai, S16 06-01

❑ <http://www.comp.nus.edu.sg/~CS5234/2002/>

✧ Course Objectives:

❑ covers *advanced* design and analysis of algorithms

❑ emphasizes efficient algorithms and data structures

❑ Algorithms for combinatorial optimisation

❑ Use of algorithms in advanced software development

✧ Target Students:

❑ advanced undergraduate (4th year) students

❑ beginning graduate students

❑ those doing research in algorithm design

CS5234: Overview...(2)

★ Prerequisite:

- ❑ **CS3230 Design and Analysis of Algorithms AND**
- ❑ **C++ Programming Skills**
- ❑ **Or Consent of Instructor**

★ TextBook and Reference Material:

- ❑ **[CLRS01] *Introduction to Algorithms*, (2nd edition)
by Cormen, Leiserson, Rivest, Stein [COOP, @\$7x]**
- ❑ **(Material also comes from [Tarj83], [RND77], [SKD83])**

CS5234: Overview... (3)

★ Course Grading:

- ❑ **30% Homeworks**
- ❑ **20% Programming Assignments**
- ❑ **50% Final Exam (OPEN BOOK)**

★ Homework: (30%)

- ❑ **About 4-5 sets (about 1 every 2-3 weeks)**
- ❑ **Very important part of the course**

★ Programming Assignments: (20%)

- ❑ **Two C++ programming assignments**
- ❑ **Uses LEDA Library (C++)**

CS5234: Topics (Tentative)

★ ADVANCED DATA STRUCTURES

- ❑ *Heaps, Shortest Path and MST*
- ❑ *Leftist Heaps and $O(m \lg \lg n)$ MST Alg.*
- ❑ *Amortized Analysis and Binomial Heaps [Fast Review]*
- ❑ *F-Heaps & Fast Shortest Path Alg.*

★ COMBINATORIAL Optimization

- ❑ *Maximum Matching: Algorithms and Applications*
- ❑ *Network Flow Problems: Algorithms and Applications*

★ NP-COMPLETENESS

- ❑ *NP-completeness: Cook's Theorem, Reduction*
- ❑ *Approximation Algorithms*
- ❑ *Local Search Methods*

★ SPECIAL TOPICS:

- ❑ *TSP & TTP, Knapsack Problem, Graph Partitioning*

CS4534: References

✧ Main Text:

[CLRS01] *Introduction to Algorithms, (2nd edition)* (SC-RBRx2)
by Cormen, Leiserson, Rivest, Stein, MIT, (2001)

✧ References:

[Tar83]* *Data Structures and Network Algorithms,* (SC-RBRx1)
by Robert E. Tarjan, SIAM, 1983.

[RND77]* *Combinatorial Algorithms: Theory & Practice,* (SC-MSx1)
by Reingold, Nievergelt & Deo, PH, (1977)

[SKD83]* *Discrete Optimization Alg. with Pascal Programs,* (SC-MSx2)
by Syslo, Kowalik & Deo, PH, (1983)

[PS82] *Combinatorial Optimization: Alg. & Complexity,* (SC-RBRx1)
by Papadimitriou & Steiglitz, PH, (1982)

About CS5234 Homeworks

★ RSA Problem

- ❑ **Routine Problems -- easy practice problems**
- ❑ **Standard Problems -- to be submitted**
- ❑ **Advanced Problems -- for challenge, fun**

About CS5234 Homeworks – (2)

★ Academic Policy (on Plagiarism)

- ❑ You do your work **YOURSELF**
- ❑ If **REALLY** stuck, discuss **ONLY** general approaches
 - ◆ *IF YOU DISCUSS, write names of collaborators*
- ❑ Do **NOT** copy/compare answers!

★ Your Homework Answers:

- ❑ **Concise & Precise Answers**
- ❑ **Appropriate Level of Detail (see handouts)**

CS5234: Is this the right Course for You?

- ★ I ASSUME knowledge of
 - ❑ Programming (CS1101); Data Structures (CS1102);
 - ❑ AND CS3230 Analysis of Algorithms;
- ★ If you have NOT done CS3230 or equiv.:
 - ❑ Take CS3230 instead!

This is NOT a course to learn algorithms?

- ❑ We assume that you ALREADY KNOW algorithms
- ❑ TAKE CS3230 instead!

CS5234 Pre-Requisites...

- ✧ **CS3230 Analysis of Algorithms**

- ✧ **Else, Background Knowledge Assumed**

- Alternatively, see that you have *mastered* the following Chapters from the text [CLRS01] (including the analysis of their running times):

- ◆ *Chapters 3-5* *Mathematical Foundations*
- ◆ *Chapters 6-13* *Basic Data Structures*
- ◆ *Chapters 22-24 Graph Algorithms*

- ✧ **Otherwise:**

- Talk to instructor (case-by-case)**

Else, Background Knowledge Assumed....

❑ Data Structures: (with Analysis)

- ◆ *Stacks, Queues, Lists,*
- ◆ *Binary search trees, balanced trees,*
- ◆ *Heaps and priority queues*

❑ Algorithm Design Paradigms (with Analysis)

- ◆ *Standard sorting and searching algorithms*
- ◆ *Graph algorithms: DFS, BFS,*
- ◆ *Dijkstra's SP alg, MST Algorithms*
- ◆ *Greedy Algorithms, Dynamic Prog, Divide-and-Conquer*

❑ Analysis of Algorithms

- ◆ *Expertise with Big-O notations*
- ◆ *Competent with Algorithm Analysis:*
 - * Quicksort, DFS, BFS, Divide-and-Conquer alg.

Q & A.

--- The End ---