

CS3340 Analysis of Algorithms

Instructor Dr. Kaizhong Zhang

Office MC 372

Tel. Ext. 83826

Email [kzhang<at>csd.uwo.ca](mailto:kzhang@csd.uwo.ca)

Home Page www.csd.uwo.ca/courses/CS3340b/

- **Textbook:**

Introduction to Algorithms (third edition, 2009)

by T.H. Corman, C.E. Leiserson, R.L. Rivest, and C. Stein

- **Assignments:**

3 assignments, 10 % each

all three will involve some programming

use Java, C or C++

should be independent work

- **Examinations:**

Midterm Exam, 25%

Final Exam, 45%

close-book exams

Goals of the course

- Survey important data structures and algorithms to help us design efficient programs (software)
- Introduce mathematical techniques for the analysis of algorithms

Synopsis

- **Data structures:** methods of organizing data involved in computation
- **Algorithms:** precisely stated general problem-solving methods suitable for computer implementation
- They are central objects of study in computer science
- They go hand-in-hand: neither can be studied fruitfully without knowledge of the other

World of algorithms

- Sequential algorithms
- Parallel algorithms:
many computers or processes working concurrently
 - synchronously: all computers working together to solve a problem such as sorting
 - asynchronously: computers working independently usually on a network (distributed database systems)
- Randomizing algorithms:
flip a coin, use outcome wisely
- Approximation algorithms:
find good approximation in polynomial time

(... continued)

- Computational geometry:
very useful for robotics
- Computational biology algorithms:
solving molecular biology problems with computational methods
- Genetic algorithms:
use mutation, splicing and other genetic principles for optimization
- DNA computing algorithms:
used for computation by solely manipulating DNA strands
- etc.

Topics

- Induction, order of magnitude, solving recurrence relations.
- Binary trees. Basic set operations (search, insert, delete, intersection, union). Heaps.
- red-and-black trees: simplest B-tree.
- Sorting. Various sorting methods: quicksort, mergesort, heapsort, etc. Lower bounds on sorting. Selection.
- String matching and sequence comparison. Huffman coding.
- Union-find.
- Graph algorithms:
depth-first search, cycles, topological sort, shortest paths, transitive closure, spanning trees, connected components, maximum flow.

(... continued)

- Algorithms design techniques:
divide-and-conquer, dynamic programming, analysis of recurrences
- NP-completeness
- Parallel algorithms