CS251—Data Structures

(`http://www.cs.purdue.edu/homes/spa/cs251.html`)

MWF 10:30–11:20 in *UNIV 303* and 1:30–2:20 in *CS GO 66*

Professor:	W. Szpankowski
E-mail:	`spa@cs.purdue.edu` (can be used *only* in emergency)
Office:	cs144, 49-46703
Office Hours:	WF 3:30–4:30 or by appointemt
T.A.:	
Office:	
Office Hours:	

Texts:

[S]: R. Sedgewick *Algorithms in C*: Parts 1–4 (Third Edition), Addison Wesley (1998), ISBN 0-201-31452-5.

[A]: A. V. Aho, J. E. Hopcroft and J. D. Ullman, *Data Structures and Algorithms*, Addison–Wesley (1983), ISBN 0–201–00023–7.

E. Horowitz, S. Sahni, and S. Anderson-Freed, *Fundamentals of Data Structures in C* , Computer Science Press (1993), ISBN 0-7167-8250-2.

Description

Algorithms will be presented in a Pseudo-code or C. Projects must be written in Cor C++. We will try to cover the following topics ([A] refers to Aho&Hopcroft&Ullman book while [S] denotes the Sedgewick book):

- algorithms and their analysis—big O, Ω, proof by induction [A]: [**Ch 1.4–1.6, 9**]; [S]: 2.1-2.7.
- primitive data types, abstract data types, data structures [A]: [**Ch 1.2, 1.3**]; [S]: 4.1.
- single and multidimensional arrays, triangular arrays and variations, strings [A]: [**Ch 2.5**]; [S]: 3.2.
- stacks, removing recursion, queues, memory management for multiple stacks [A]: [**Ch 2.3, 2.4, 2.6**]; [S]: 4.2-4.6.
- singly linked lists, circular lists, doubly linked lists [A]: [**Ch 2.2**]; [S]: 3.3-3.4.
- trees—general trees, priority queues, ninary search trees, 2-3 trees and height-balanced trees, tries [A]: [**Ch 3, 4.10, 5.1, 5.2, 5.4, 5.3**]; [S]: 5.4-5.7, 9.1-9.6, 12.1-12.6, 13.3, 15.1-15.4.
- sets, union/find algorithm [A]: [**Ch 4.1–4.4, 5.5, 5.6**]
- hashing [A]: [**Ch. 4.7**]; [S]: 14.1-14.5.
- graphs and digraphs [A]: [**Ch 6, 7**]; [S]: 5.8.
- Sorting and Searching [A]: [**Ch 8**]; [S]: 6.1-6.4, 7.1-7.2, 8.1-8.2.
- memory management of fixed and variable size records [A]: [**Ch 12.1, 12.2**]; [S]: 3.5
- recurrence equations [A]: [**Ch 9.3**]; [S]: 2.5.

Course Work and Grading Policy

The course work consists of homeworks, programming projects, a midterm and a final. The final grade is based on 35% homeworks and programming projects, 30% midterm and 35% final.

No late assignments will be accepted. No incompletes. If you want to re-grade your homework or project, please contact your TA within 10 days of receiving the homework back. No re-grading after this time.