

# CST 280 Computer Science III / Data Structures (Fall 2009)

## Time and location

- CST 280-10: Tuesday, 6:00 pm - 8:30 pm, Chicago Campus, GB 310G.
- CST 280-24: Wednesday, 6:30 pm - 9:00 pm, Schaumburg Campus, SCH 712.

## Instructor

Dr. Evgeny Dantsin

- Email: [edantsin@roosevelt.edu](mailto:edantsin@roosevelt.edu)
- Webpage: [cs.roosevelt.edu/~dantsin/](http://cs.roosevelt.edu/~dantsin/)
- Office hours:
  - Tuesday, 4:30 pm - 6:00 pm, Chicago Campus, GB 506B
  - Thursday, 5:00 pm - 6:30 pm, Schaumburg Campus, SCH 600I

## Course description

The course focuses on basic techniques used to design and analyze efficient algorithms. The major topics include: data structures for sets, searching algorithms, divide and conquer, sorting algorithms, greedy algorithms, dynamic programming, pattern matching and fingerprinting, computational complexity.

## Textbook

Richard Johnsonbaugh and Marcus Schaefer, *Algorithms*, Pearson Prentice Hall, 2004. ISBN: 0023606924.

## Prerequisites

- MATH 245 (Discrete Structures)
- CST 250 (Computer Science II)

## Tentative schedule

Week	Topics	Homework	Textbook
1	Class introduction. Background for the course: mathematics for algorithms.		1.1-1.4, 2.1-2.3, 2.5-2.6
2	Data structures for sets. Abstract data types. Stacks, queues, linked lists. Graph representation.	Project-1	3.1-3.4
3	Binary search. Recursive relations and main recurrence theorem. Depth-first search. Breadth-first search.		2.4, 4.1-4.3
4	Topological sort. Backtracking algorithms. Divide and conquer. Mergesort. Integer multiplication. Matrix multiplication.	HW-1	4.4-4.5, 5.1-5.4
5	Sorting algorithms. Insertion sort. Quicksort. Randomized quicksort. Lower bound for comparison-based sorting.		6.1-6.3
6	Linear-time sorting algorithms. Counting sort and radix sort. Selection. Description of Project-2.	Project-2	6.4-6.4
7	Midterm exam.		
8	Optimization problems and greedy algorithms. Examples: coin		7.1

	changing, shortest paths, scheduling, set cover.		
9	Greedy algorithms on graphs. Data structures: disjoint sets, priority queues implemented with heaps.	HW-2	7.2-7.3, 3.5-3.6
10	The shortest-paths problem and Dijkstra's algorithm. Data compression and Huffman's algorithm. The knapsack problem.		7.4-7.6
11	Dynamic programming. Computation of Fibonacci numbers. Spelling correction and the edit distance.	HW-3	8.1, 9.5
12	Pattern matching. Fingerprinting and the Rabin-Karp algorithm for text searching. Monte Carlo and Las Vegas versions.		9.1-9.2
13	Polynomial-time algorithms. The class P. Polynomial-time verifiers and the class NP.	HW-4	10.1-10.2
14	Review of the course. Preparation for the final exam.		
15	Final exam.		

## Assignments and grading

Grades will be determined by the total number of points earned on the following assignments:

- two projects (mandatory for passing the course);
- homework assignments and in-class quizzes (total maximum: 60 points);
- midterm and final exams (20 points each).

No late homework will be accepted. No extra credit work will be assigned and no make-ups will be given. The exams are open-book and open-note. Instances of academic dishonesty will be handled as described in University policies. Grades will be assigned according to the following scale:

Grade	Interval
A	$\geq 90$ pts
A-	$\geq 85$ pts and $< 90$ pts
B+	$\geq 80$ pts and $< 85$ pts
B	$\geq 75$ pts and $< 80$ pts
B-	$\geq 70$ pts and $< 75$ pts
C+	$\geq 65$ pts and $< 70$ pts
C	$\geq 60$ pts and $< 65$ pts
C-	$\geq 55$ pts and $< 60$ pts
D+	$\geq 50$ pts and $< 55$ pts
D	$\geq 45$ pts and $< 50$ pts
D-	$\geq 40$ pts and $< 45$ pts
F	$< 40$ pts

## Course website

Lecture notes, slides, homework assignments, and other course materials will be posted on [Blackboard](#) after every lecture.