CENG 201 Data Structures and Algorithms

Summer 2022

Instructor:	Hasan Erbay	**Time:**	Monday 8:40 – 11:30 Thursday 8:40 – 11:30
Email:	herbay@thk.edu.tr	**Place:**	201

Course Pages: http://moodle.thk.edu.tr/

Office Hours: By appointment.

Main References:

- Problem Solving With Algorithms and Data Structures Using Python by Brad Miller and David Ranum, Luther College It can be reached at: https://runestone.academy/runestone/books/published/pythonds/index.html
- Lecture notes.

Objectives: Achieve an understanding of fundamental data structures and algorithms and the tradeoffs between different implementations of these abstractions. Theoretical analysis, implementations, and applications. Lists, stacks, queues, heaps, dictionaries, maps, hashing, trees and balanced trees, sets, and graphs, searching and sorting algorithms are included.

Learning outcome: Students will be able to:

- Describe, explain, and use abstract data types including stacks, queues, lists, sets, maps and graphs.
- Implement those data types using both contiguous and linked representations.
- Implement a variety of algorithms for searching and sorting, including linear search, binary search, insertion sort, selection sort, merge sort, quicksort, and heap sort.
- Read and write recursive algorithms. Understand when recursion is, and is not, appropriate.
- Analyze the time and space efficiency of data structures and algorithms and apply this analysis to select the best tools for solving particular problems.

  analyse the runtime performance of a (simple) algorithm/program in terms of the size of its inputs, and this in the average, best, and worst cases. choose appropriate algorithms and data structures for storing data, searching and sorting, as well as implement those algorithms. use and implement basic graph algorithms.

Prerequisites: An undergraduate-level understanding of probability, statistics, algorithms, and linear algebra is assumed.

Tentative Course Outline:

Week	Content	Duration (Hours)	Deliver
1	Course Logistics, Learning Objectives, Grading	3	Face-to-face/Online
2	Linear data structures: Stacks, queues and lists	3	Face-to-face/Online
3	Introduction to complexity: $\mathcal{O}()$-notation	3	Face-to-face
4	Trees, binary trees	3	Face-to-face
5	Balanced trees, AVL trees heaps, priority queues	3	Face-to-face
6	Heaps, priority queues, hashing	3	Face-to-face
7	Collusion and resolution strategies, double hashing	3	Face-to-face
8	Midterm	3	Face-to-face
9	Sorting algorithms,	3	Face-to-face
10	Recurrences, mergesort	3	Face-to-face
11	Graphs, DFS	3	Face-to-face
12	BFS, Weighted graphs, Djikstra's	3	Face-to-face
13	Minimum Spanning Trees	3	Face-to-face
14	Kruskal's Algorithm, Prim's Algorithm	3	Face-to-face

Grading Policy: Midterm (40%), Final (60%).

Class Policy:

- Regular attendance is essential and expected.