

| | |
|--------------------|-----------------|
| BLOCK NAME | DATA STRUCTURES |
| BLOCK CODE | CS-L1B3 |
| COURSE | 1 |
| LEVEL | 1 |
| CREDITS | 6 |
| CLASS HOURS | 60 |
| HOMEWORK | 90 |
| TOTAL HOURS | 150 |

DESCRIPTION

This block introduces the most common data structures a developer usually uses when creating software. We will face practical challenges that will facilitate the learning of the operation of the most common data structures (lists, trees, graphs and hash tables). For this to be done, we first create our own implementation for such data types and then we integrate our libraries into a program we created for a previous block.

PRE-REQUISITES

Basic programming skills are needed.
CS-L1B1

OBJECTIVES

The goal is for students to know the most common data structures and their advantages/disadvantages. This will enable students to choose the correct one for each case.

SKILLS TO BE DEVELOPED

- 1 - Lists.**
 - 1.1 - Use python lists.
 - 1.2 - Use python lists as stacks.
 - 1.3 - Use python lists as queues.
 - 1.4 - Be able to write your own implementation of lists using arrays.
 - 1.5 - Be able to write your own implementation of linked lists.
 - 1.6 - Understand the advantages and drawbacks of lists, stacks and queues and know when to use each one depending on the problem to solve.
- 2 - Trees.**
 - 2.1 - Be able to write your own implementation of a binary search tree datatype.
 - 2.2 - Understand the advantages and drawbacks of binary search trees and know when to use them depending on the problem to solve.
- 3 - Graphs.**
 - 3.1 - Be able to write your own implementation of a graph datatype.
 - 3.2 - Understand the advantages and drawbacks of graphs and know when to use them depending on the problem to solve.
- 4 - Hash tables.**
 - 4.1 - Be able to write your own implementation of a hash table.
 - 4.2 - Understand the advantages and drawbacks of hash tables and know when to use them depending on the problem to solve.
- 5 - Global lessons.**
 - 5.1 - Understand how data structures differ from basic data types.
 - 5.2 - Calculate computational complexity and memory complexity of choosing the correct data structure for a problem, knowing the advantages and disadvantages of each one

SYLLABUS

- 1 - Lists.
- 2 - Trees.
- 3 - Graphs.
- 4 - Hash tables.

METHODOLOGY

Resolution of practical activities supervised by the mentor. Compulsory attendance.

DEDICATION AND EVALUATION

The student must pass the mandatory activities (challenges/projects) that are covered in the block.

Each challenge/project produces its own score and has been designed to cover certain block percentages.

Such score is 80% objective (the program that solves the challenge/project works without errors and producing the expected results) and 20% subjective (solution elegance, how clean the code is, documentation).

Block scores are finally calculated by prorating individual activities with respect to their block coverage percentages.