Course «Algorithms and Data Structures»

6B06102 – Information Systems
6B06109 – Artificial Intelligence
6B06103 – Computer Engineering
6B06104 – Computer Science·



L1. Introduction to Algorithms and Data Structures

Acting Professor, Cand. Phys.-Math. Sc. Ualiyeva Irina Maratovna

Today

Part I · Course logistics

How the course works, what we study, how you are graded

Part II · Introduction

What algorithms and data structures are, and why they matter

You already write programs. This course is about what makes a program fast.

What we'll study: 15 weeks

Week	Topic
1	Introduction to algorithms and data structures
2	Pseudocode and flowcharts
3	Time complexity: Big-O
4	Arrays; sequential and binary search
5	Recursion
6	Linked lists
7	Stack and queue
8	Midterm

Week	Topic
9	Binary trees and BST
10	Simple sorting
11	Merge sort, quick sort
12	Hash tables
13	Graphs: representation, BFS
14	Graphs: DFS, applications
15	Review

Grading and rules

Component	Weight
Labs and assignments	6
Quizzes / class work	36
Midterm (week 8)	18
Final exam	40

- **Instructor:** i.ualiyeva@gmail.com

How to succeed in this course

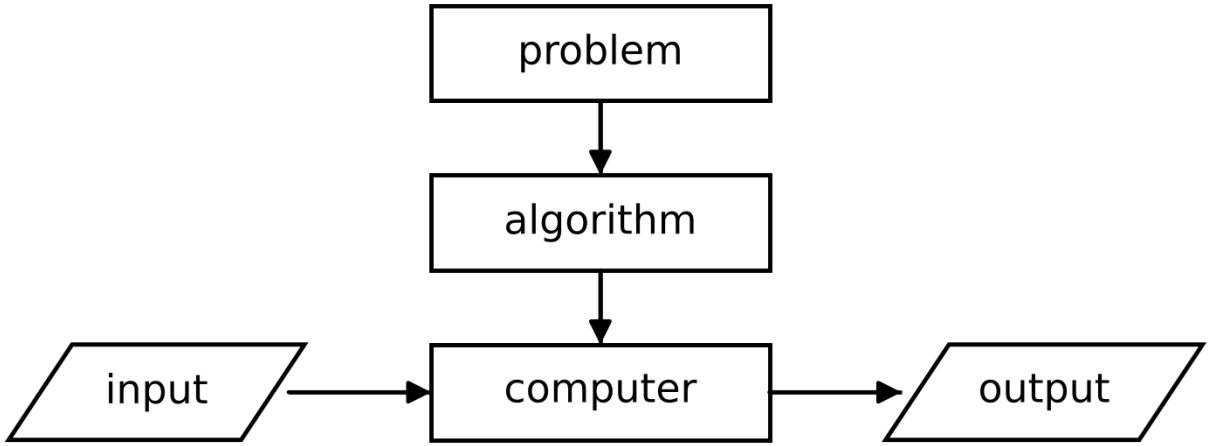
Algorithms are learned with a pencil.

When you see an algorithm, take a small example and run it by hand, step by step. If you can’t trace it on paper, coding it won’t help.

Trace it, then code it, then test it. Every week.

What is an algorithm?

An **algorithm** is a sequence of unambiguous instructions for solving a problem, i.e., for obtaining a required output for any legitimate input in a finite amount of time.



Five properties of every algorithm

Property	Meaning
Finiteness	it always stops after a finite number of steps
Definiteness	every step is precise and means one thing
Input	zero or more inputs from a given set
Output	one or more outputs related to the input
Effectiveness	every step can be done exactly, even with pencil and paper

Donald Knuth, one of the founders of computer science

Why the choice of algorithm matters

Find one name among **1,000,000** names sorted alphabetically.

1,000,000

comparisons: sequential search, one by one

20

comparisons: binary search, halve the book each time

Each comparison halves the book, so we need $\log_2 10^6 \approx 20$ steps.

Binary search works only because the names are **sorted**.

A faster computer won’t save a bad algorithm

Running time at 10^9 operations per second:

n	n operations	n^2 operations	2^n operations
10	instant	instant	instant
1,000	instant	0.001 s	longer than the age of the universe
1,000,000	0.001 s	about 17 minutes	—

In week 3 we learn to describe this growth with Big-O.

From a problem to a program

1. **Understand** the problem; try small examples and special cases
2. **Choose** the approach and the data structure
3. **Design** the algorithm and write it in pseudocode
4. **Prove** that it is correct for every legitimate input
5. **Analyze** its time and memory
6. Only then: **code** and test

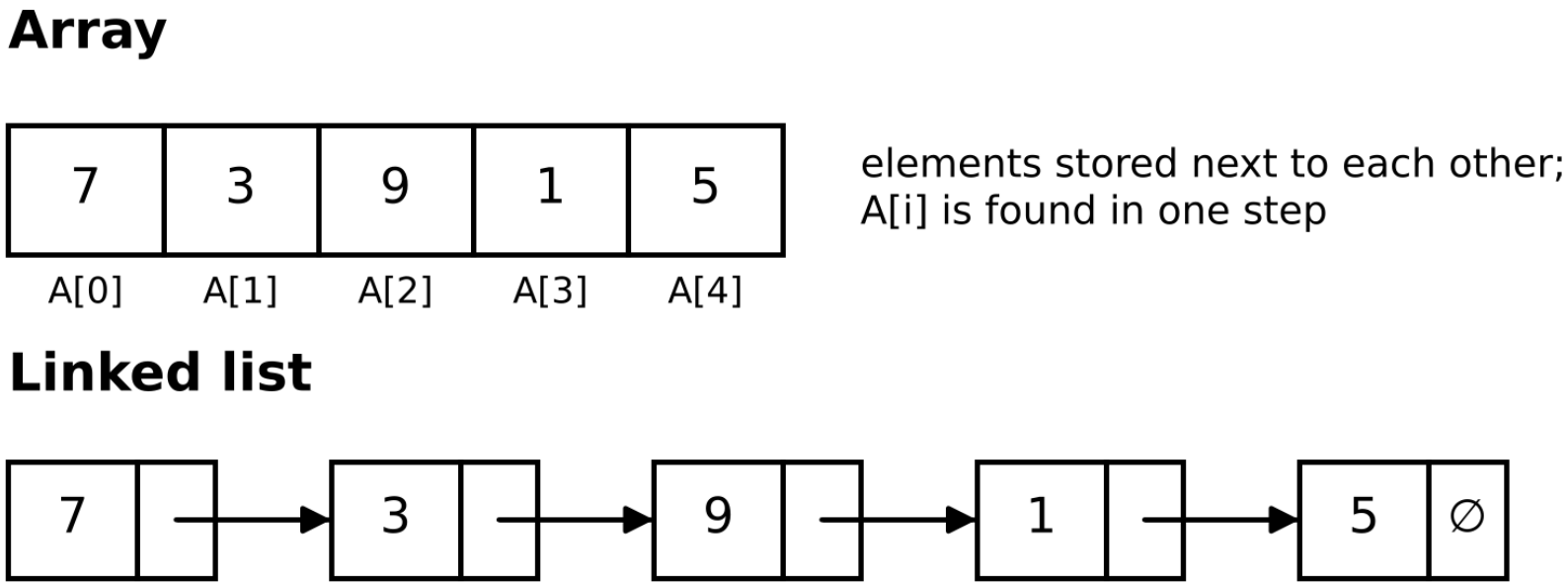
Most beginners jump straight to step 6.

Important problem types

Problem type	What it means	In this course
Sorting	arrange items in order	weeks 10–11
Searching	find an item with a given key	weeks 4, 9, 12
String processing	search in text, compare strings	later courses
Graph problems	routes, networks, connections	weeks 13–14
Combinatorial	best arrangement or subset	later courses
Geometric	points, lines, polygons	later courses
Numerical	equations, computations with real numbers	later courses

What is a data structure?

A **data structure** is a way to organize data. It decides which operations are fast.



Array: A[i] in one step, but inserting in the middle is slow. Linked list: insertion is easy, but finding the i-th element is slow.

Data structures in this course

Linear

arrays, linked lists, stacks, queues

Trees

hierarchies; binary search trees

Hash tables

very fast lookup by key

Graphs

vertices connected by edges

An **abstract data type** is a set of operations (for example push and pop for a stack) without fixing how the data is stored.

Summary

- An algorithm is a clear, finite set of steps that solves a problem for every allowed input.
- A good algorithm can be thousands of times faster than a bad one.
- A data structure is a way to organize data; it decides which operations are fast.
- Think first, code last.

Review questions

1. What is an algorithm? Explain the definition in your own words.
2. Why is binary search much faster than sequential search? What must be true about the data before we can use binary search?
3. What is a data structure? Name three data structures we will study in this course.

Reading: Levitin, Sections 1.1, 1.3, 1.4; Cormen et al., Chapter 1.