



哈爾濱工業大學(深圳)
HARBIN INSTITUTE OF TECHNOLOGY, SHENZHEN

数据结构 Data Structures

Chapter 1 Introduction to Data Structures

Prof. Yitian Shao
School of Computer Science and Technology

Instructor Information

Prof. Dr. **Yitian Shao** (Chinese: 邵奕天)

- School of Computer Science and Technology
- Research interests: **haptic interfaces**, **robotic tactile sensing**, and **wearable technologies**

Academic Background

- Junior professor (W1) of Electrical and Computer Engineering, TU Dresden, Germany (2022-2023)
- Postdoctoral Researcher, Max Planck Institute for Intelligent Systems, Stuttgart, Germany (2021-2022)
- Ph.D., Electrical and Computer Engineering, UC Santa Barbara, U.S. (2015-2020)



Email: shaoyitian@hit.edu.cn (I also teach High-level Language Programming)

Teaching Assistant

Lin Xu

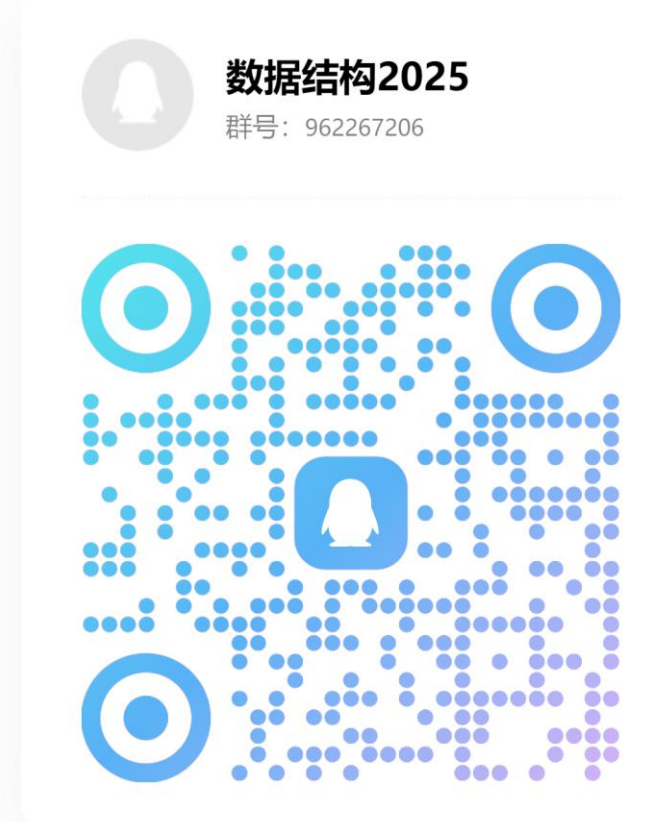
School of Computer Science and Technology

Grading **Quiz** and **Lab** submissions

Email: sjjg2025@gmail.com

Let's Get Connected!

- Course related discussion: **QQ**
 - Posting homework
 - Sharing course materials
 - Course related Q&A
- **Step 1:** Download and install QQ on your mobile/PC: im.qq.com/index



- **Step 2:** Inside your QQ, scan the QR code above

Overview

- In *Data Structures*, we learn:
 - Basic **abstract data types** and their **logical structures, storage structures** and **operations**
 - Corresponding **algorithms** and **typical applications**
 - The storage and application of some **advanced data structures**
 - Common **searching** and **sorting** algorithms
- Course objectives:
 - Understand the relationship between the data structures and algorithms
 - Properly use data structures and algorithms
 - Analyze code in time complexity (and space complexity)
 - Design effective algorithms and data structures to solve practical problems

Expectations

- **Attend lectures on time**
 - **Complete all assignments on your own** and **submit on time**
 - Engage in class discussion
 - Communicate with instructor if need help
-
- All lectures are given in **English**
 - All assignments and exams must be completed in **English**

Academic Integrity

Quiz

Lab

Final Exam

- **Do not** look at other people's solution code
- **Do not** give your solution code to others, or post it elsewhere
- **Report** any inappropriate activity you see performed by others

All assignments are checked for similarity! **Copied solutions suffer significant grade penalty!**

Talk to the instructors/TA if you need help with the assignment (but **do not wait until the quiz or exam day**)

Grading

- 24 lectures from Week 1 to Week 13
 - **Tuesday 2pm-3:45pm** and **Thursday 10:30am-12:15pm**, T5-507
- Course assignments and exams
 - **Exercises** will not be graded! (Complete them if you want to prepare yourself for the quiz and final exam)
 - **In-class quiz** $\times 4 = 40\%$ points (**Complete and Submit in class**)
 - **Lab report** $\times 4 = 20\%$ points (**Complete and Submit in lab**)
 - **Final exam** = **40% points**

Late submission not accepted, no excuse!

Special cases

- **Require a valid proof** for late submission / Absenteeism
- **Must inform the instructor** before taking any actions
- **Submit a valid proof** no later than a week

Valid Stamp →

北京同仁医院
诊断证明书 病案号 572
姓名 [Your Name] 性别 男 年龄 53 科别 门诊
诊断及建议: 右眼RB. 右眼对光反射弱.
(盖章有效)
医师 [Signature]
2010年8月20日
学字245

The image shows a medical certificate from Beijing Tongren Hospital. It features a red circular stamp in the center with the text '北京市同仁医院 眼科 诊断证明专用章 (S4)'. A red arrow points from the text 'Valid Stamp' to this stamp. At the bottom right, there is a signature of a doctor, and another red arrow points from the text 'Valid Signature' to this signature. The form includes fields for patient name, gender, age, department, and a section for diagnosis and advice. The date is 2010年8月20日, and the certificate number is 学字245.

← **Valid Signature**

Class Attendance

- **Class attendance is enforced** and recorded every single class
- Bring a laptop to the class if possible
- **If you miss more than 8 lectures, you will be disqualified to attend the final exam**

Class Attendance

- **Class attendance is enforced** by our university
- Technical issues – **Get help from H719** (Main building)
- Install **WeChat** or the **Attendance Check APP**

安卓 Android



苹果 IOS



Class Attendance Check!

- Technical issues – **Get help from H719 (Main building)**
- Open **WeChat** or the **Attendance Check APP**
- Now, **scan the QR code** using WeChat or Attendance Check APP, ask for help if you encounter any problem.

Schedule

Tue

Wed

Thu

Week 1: Introduction (2/25)

Week 2: Array and Matrix (3/4)

Week 3: Linked Lists I (3/11)

Week 4: Stacks and Queue I (3/18)

Week 5: Strings I (3/25)

Week 6: Trees I (4/1)

Week 7: Trees III (4/8)

Week 8: Trees V (**4/15 Quiz Day!**)

Week 9: Graph II (4/22)

Week 10: Searching Algorithms I (4/29)

Week 11: Searching Algorithms II (5/6)

Week 12: Sorting Algorithms I (5/13)

Week 13: Sorting Algorithms I (5/20)

Lab 1

Lab 2

Lab 3

Lab 4

Review of C++ Programming (2/27)

Basics of Algorithm (3/6)

Linked Lists II (**3/13 Quiz Day!**)

Stacks and Queue II (3/20)

Strings II (**3/27 Quiz Day!**)

Trees II (4/3)

Trees IV (4/10)

Graph I (4/17)

Graph III (4/24)

(holiday)

Searching Algorithms III (**5/8 Quiz Day!**)

Sorting Algorithms I (5/15)

Final Exam (Date to be determined)

Textbook and Online Resources

Eric Roberts, *Programming Abstractions in C++*, Pearson. 2013. (Electronic version acceptable)

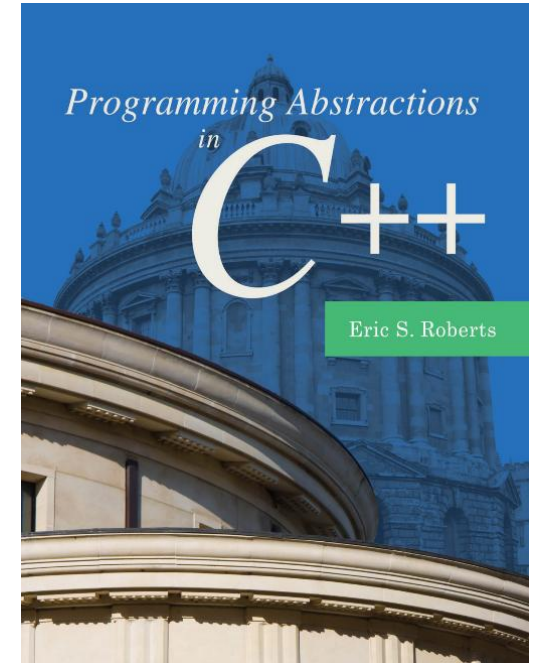
Online Course: [Stanford CS106B](#)

Copyright (C) Stanford Computer Science and Tyler Conklin, licensed under Creative Commons Attribution 2.5 License. All rights reserved.

(The content of this course is developed based on it)

Other online references and materials

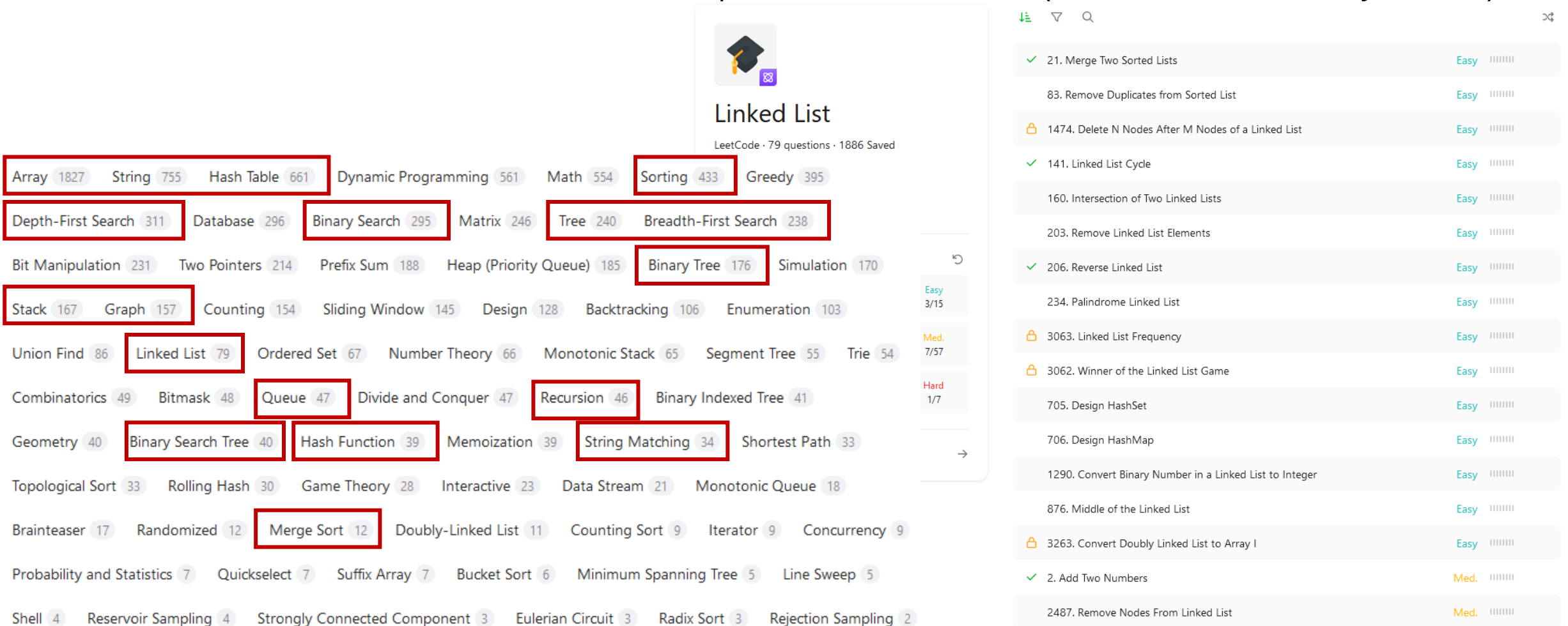
- Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, Introduction to Algorithms (3rd Edition), The MIT Press. 2009.
- Robert Sedgewick, Kevin Wayne, Algorithms (4th Edition), Addison-Wesley Professional. 2011.
- leetcode.com



LeetCode for Homework and Self Practice

leetcode.com/problemset/algorithms/

- 3431 Problems – can cover all topics in this course (start from the easy ones!)



The screenshot displays the LeetCode website interface. At the top, the 'Linked List' problem set is highlighted with a red box. Below it, a list of algorithms is shown, each with a problem count. The 'Merge Sort' problem is highlighted with a red box. The right side of the image shows a list of problems, including '21. Merge Two Sorted Lists', '83. Remove Duplicates from Sorted List', '1474. Delete N Nodes After M Nodes of a Linked List', '141. Linked List Cycle', '160. Intersection of Two Linked Lists', '203. Remove Linked List Elements', '206. Reverse Linked List', '234. Palindrome Linked List', '3063. Linked List Frequency', '3062. Winner of the Linked List Game', '705. Design HashSet', '706. Design HashMap', '1290. Convert Binary Number in a Linked List to Integer', '876. Middle of the Linked List', '3263. Convert Doubly Linked List to Array I', '2. Add Two Numbers', and '2487. Remove Nodes From Linked List'.

Linked List
LeetCode · 79 questions · 1886 Saved

Array 1827 String 755 Hash Table 661 Dynamic Programming 561 Math 554 Sorting 433 Greedy 395

Depth-First Search 311 Database 296 Binary Search 295 Matrix 246 Tree 240 Breadth-First Search 238

Bit Manipulation 231 Two Pointers 214 Prefix Sum 188 Heap (Priority Queue) 185 Binary Tree 176 Simulation 170

Stack 167 Graph 157 Counting 154 Sliding Window 145 Design 128 Backtracking 106 Enumeration 103

Union Find 86 Linked List 79 Ordered Set 67 Number Theory 66 Monotonic Stack 65 Segment Tree 55 Trie 54

Combinatorics 49 Bitmask 48 Queue 47 Divide and Conquer 47 Recursion 46 Binary Indexed Tree 41

Geometry 40 Binary Search Tree 40 Hash Function 39 Memoization 39 String Matching 34 Shortest Path 33

Topological Sort 33 Rolling Hash 30 Game Theory 28 Interactive 23 Data Stream 21 Monotonic Queue 18

Brainteaser 17 Randomized 12 Merge Sort 12 Doubly-Linked List 11 Counting Sort 9 Iterator 9 Concurrency 9

Probability and Statistics 7 Quickselect 7 Suffix Array 7 Bucket Sort 6 Minimum Spanning Tree 5 Line Sweep 5

Shell 4 Reservoir Sampling 4 Strongly Connected Component 3 Eulerian Circuit 3 Radix Sort 3 Rejection Sampling 2

21. Merge Two Sorted Lists Easy

83. Remove Duplicates from Sorted List Easy

1474. Delete N Nodes After M Nodes of a Linked List Easy

141. Linked List Cycle Easy

160. Intersection of Two Linked Lists Easy

203. Remove Linked List Elements Easy

206. Reverse Linked List Easy

234. Palindrome Linked List Easy

3063. Linked List Frequency Easy

3062. Winner of the Linked List Game Easy

705. Design HashSet Easy

706. Design HashMap Easy

1290. Convert Binary Number in a Linked List to Integer Easy

876. Middle of the Linked List Easy

3263. Convert Doubly Linked List to Array I Easy

2. Add Two Numbers Med.

2487. Remove Nodes From Linked List Med.

LeetCode: How to Use

Description Editorial Solutions Submissions

21. Merge Two Sorted Lists

Solved

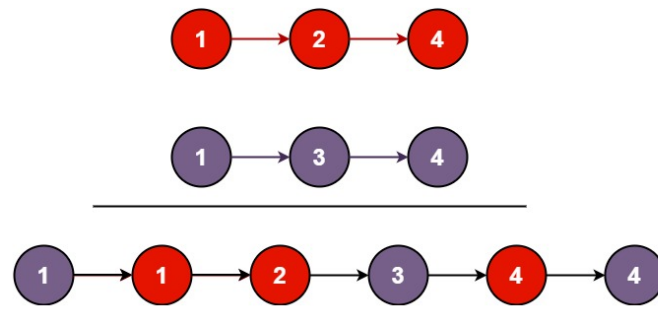
Easy Topics Companies

You are given the heads of two sorted linked lists `list1` and `list2`.

Merge the two lists into one **sorted** list. The list should be made by splicing together the nodes of the first two lists.

Return *the head of the merged linked list*.

Example 1:



Input: list1 = [1,2,4], list2 = [1,3,4]
Output: [1,1,2,3,4,4]

Example 2:

Input: list1 = [], list2 = []
Output: []

Example 3:

Input: list1 = [], list2 = [0]
Output: [0]

Constraints:

- The number of nodes in both lists is in the range `[0, 50]`.
- `-100 <= Node.val <= 100`
- Both `list1` and `list2` are sorted in **non-decreasing** order.

Seen this question in a real interview before? 1/5

Vec NaN

Run Submit

</> Code

C++ Auto

```
1 /**
2  * Definition for singly-linked list.
3  * struct ListNode {
4  *     int val;
5  *     ListNode *next;
6  *     ListNode() : val(0), next(nullptr) {}
7  *     ListNode(int x) : val(x), next(nullptr) {}
8  *     ListNode(int x, ListNode *next) : val(x), next(next) {}
9  * };
10 */
11 class Solution {
12 public:
13     ListNode* mergeTwoLists(ListNode* list1, ListNode* list2) {
14         ListNode* dummy = new ListNode(0);
15         ListNode* op = dummy;
16
17         while (list1 && list2) {
18             if (list1->val <= list2->val) {
19                 op->next = list1;
20                 list1 = list1->next;
21             } else {
22                 op->next = list2;
23                 list2 = list2->next;
24             }
25             op = op->next;
26         }
27         op->next = list1 ? list1 : list2;
28         return dummy->next;
29     }
30 };
```

Saved

Ln 10, Col 4

Testcase Test Result

Accepted Runtime: 0 ms

Case 1 Case 2 Case 3

Input

list1 =
[1,2,4]

list2 =
[1,3,4]

Output

[1,1,2,3,4,4]

Expected

[1,1,2,3,4,4]



Abstract Data Types (ADT)

What is Abstract Data Types (ADT)?

- **Data structures** can be assembled to form hierarchies. The **atomic data types**—such as `int`, `char`, `double`, and `enumerated types`—occupy the lowest level in the hierarchy.
- To represent **more complex information**, you `combine the atomic types` to form larger structures. These larger structures can then be assembled into even larger ones ...
- Collectively, these assemblages of information into more complex types are called **data structures**

What is Abstract Data Types (ADT)?

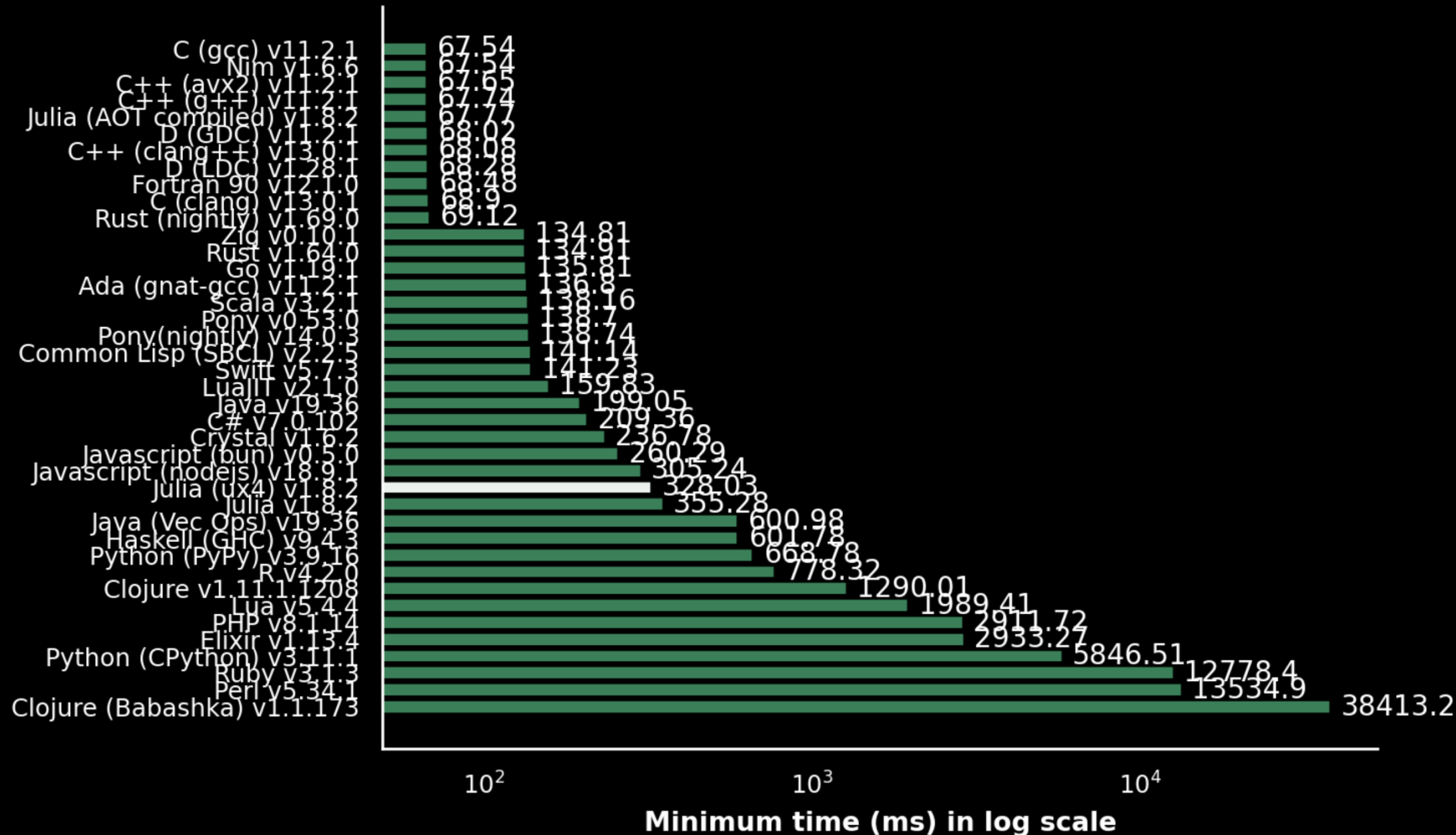
- A type defined in terms of its behavior rather than its **representation** is called an **abstract data type**, which is often abbreviated to **ADT**
- ADT are central to the **object-oriented style of programming**, which encourages **thinking about data structures in a holistic way**

Programming language for this course

- C++ (High-level Language Programming)
- Brief review of C++ syntax will be included in this course

Speed comparison of various programming languages

Method: calculating π through the Leibniz formula 100000000 times



Programming language: C++

- C++ is a programming language developed in 1983 by Bjarne Stroustrup
 - one of the world's **most widely used** languages today
 - built for systems programming with **high speed/efficiency**
 - built on older C language by adding **object-oriented programming**
 - continues to be **improved over time** (latest version: [C++ 26](#))

Basic C++ programming skills you need

- Syntax
- Compile and execute
- Variable, expression, string, **function**, **struct**, **class**

Exercise 1.1

- Homework exercises assigned after each lecture will not be graded, however, you should complete them to **get yourself prepared for quizzes and final exam!**
- Learn how to use LeetCode
- Mark your first step on LeetCode, complete [LeetCode 58](#)

58. Length of Last Word

Easy

Topics

Companies

Given a string `s` consisting of words and spaces, return *the length of the **last** word in the string.*

A **word** is a maximal [substring](#) consisting of non-space characters only.