Data Structures and Algorithms

Prepared by: Mohamed Ayman

Algorithm Engineer at Valeo

Deep Learning Researcher and Teaching Assistant

at The American University in Cairo (AUC)

spring 2020

sw.eng.MohamedAyman@gmail.com

linkedin.com/in/cs-MohamedAyman

github.com/cs-MohamedAyman

codeforces.com/profile/Mohamed_Ayman

Mohamed Ayman

Experience

- Valeo
  - Deep Learning Researcher
  - Algorithm Software Engineer
- The American University in Cairo (AUC)
  - Research Assistant
  - Teaching Assistant
- ICPC - International Collegiate Programming Contest
  - Coach at ACPC Africa and Arab Collegiate Programming Contest

Education

- MSc in Deep Learning, Cairo University
- BSc in Computer Science, Cairo University

Data Structures and Algorithms Training

Lecture Agenda

We will discuss in this lecture the following topics

1- Data Structures and Algorithms Features
2- Data Structures and Algorithms Content
3- Practice on Online Judges
4- Programming Competitions
5- Tutorials and References
6- Online Courses


Let's
START UP

Lecture Agenda

Section 1: Data Structures and Algorithms Features

Section 2: Data Structures and Algorithms Content

Section 3: Practice on Online Judges

Section 4: Programming Competitions

Section 5: Tutorials and References

Section 6: Online Courses

Data Structures and Algorithms Features

- Data structure is a way to store and organize data in order to support efficient insertions, queries, searches, updates, and deletions. Although a data structure in itself does not solve the given programming problem, the algorithm operating on it does, using the most efficient data structure for the given problem may be a difference between passing or exceeding the problem's time limit. There are many ways to organize the same data and sometimes one way is better than the other on different context.
- Algorithms is the current term of choice for a problem-solving procedure, algorithm, is commonly used nowadays for the set of rules a machine (and especially a computer) follows to achieve a particular goal. It does not always apply to computer-mediated activity, however. Algorithm is often paired with words specifying the activity for which a set of rules have been designed.
- Characteristics of Data Structures and Algorithms:

1 - Correctness: Data structures and Algorithms implementation should implement its interface correctly.

2 - Time Complexity: Running time or the execution time of operations must be as small as possible.

3 - Space Complexity: Memory usage of a data structure operation should be as little as possible.

Data Structures and Algorithms Features

- **Why we need data structures and algorithms?** there are several advantages of using them, few of them are as follows:

1. Data Organization: We need a proper way of organizing the data so that it can accessed efficiently when we need that particular data. DS provides different ways of data organization so we have options to store the data in different data structures based on the requirement.

2. Efficiency: The main reason we organize the data is to improve the efficiency. We can store the data in arrays then why do we need linked lists and other data structures? because when we need to perform several operation such as add, delete update and search on arrays , it takes more time in arrays than some of the other data structures. So the fact that we are interested in other data structures is because of the efficiency.

- **Time Complexity:** It is a way to represent the amount of time required by the program to run till its completion. It's generally a good practice to try to keep the time required minimum, so that our algorithm completes it's execution in the minimum time possible. We will study about Time Complexity in details in later sections.
- **Space Complexity:** Its the amount of memory space required by the algorithm, during the course of its execution. Space complexity must be taken seriously for multi-user systems and in situations where limited memory is available.

Data Structures and Algorithms Features

Data Structures and Algorithms Features

Lecture Agenda

- ✓ Section 1: Data Structures and Algorithms Features
- **Section 2: Data Structures and Algorithms Content**
- Section 3: Practice on Online Judges
- Section 4: Programming Competitions
- Section 5: Tutorials and References
- Section 6: Online Courses

Data Structures Content

Part 1: Linear Data Structures

Lecture 1: Complexity Analysis & Recursion

Lecture 2: Array

Lecture 3: Linked List

Lecture 4: Stack

Lecture 5: Queue

Lecture 6: Deque

Lecture 7: Built-in Linear Data Structures

Part 2: Non-Linear Data Structures

Lecture 8: Binary Tree

Lecture 9: Binary Search Tree

Lecture 10: Self Balancing Binary Search Tree

Lecture 11: Binary Heap Tree

Lecture 12: Hash Table

Lecture 13: Graph

Lecture 14: Built-in Non Linear Data Structures

© Prepared by: Mohamed Ayman

Data Structures Content

Part 3: Advanced Data Structures

Lecture 15: Disjoint Set

Lecture 16: Skip List

Lecture 17: Trie

Lecture 18: Segment Tree

Lecture 19: Binary Indexed Tree (Fenwick Tree)

Lecture 20: Treap (Randomized Binary Search Tree)

Part 4: Advanced Data Structures

Lecture 21: K-Dimensional Tree

Lecture 22: Sparse Table

Lecture 23: B/B+ Tree

Lecture 24: Suffix Array

Lecture 25: Suffix Tree

Lecture 26: Advanced Trees

Hands-on Projects & Assignments & Practices

Data Structures Projects (4 Projects)

Project 1: Mathematical Equations Calculator	(linear data structures application)
Project 2: Mobile Contacts Indexing	(linear & non-linear data structures application)
Project 3: Big Families	(non-linear data structures application)
Project 4: University Friends	(non-linear data structures application)

Data Structures Assignments (10 Assignment)

- After each lecture we have an assignment (Implementing & Testing the Data Structures on each Lecture)

Data Structures Practices (30+ Practice Problems) on each Lecture.

Algorithms Content

Part 1: Basic Algorithms

Lecture 1: Analysis of Algorithms

- Analysis Methods in Time & Space Complexity
- Master theorem - Substitution method
- Recursion tree method

Lecture 2, 3: Sorting Algorithms

- Selection Sort
- Bubble Sort
- Merge Sort
- Heap Sort
- Bitonic Sort
- Bucket Sort
- Tim Sort
- Insertion Sort
- Shell Sort
- Quick Sort
- Count Sort
- Radix Sort
- Pigeonhole Sort
- Cartesian Tree Sort

Lecture 4, 5: Searching Algorithms

- Linear Search
- Ternary Search
- Exponential Search
- Fibonacci Search
- Binary Search
- Jump Search
- Sublist Search
- Interpolation Search

Lecture 6: Divide and Conquer Algorithms

- Binary Search
- Fast Power
- Count Inversions
- Merge & Quick Sort
- Closest Pair of Points
- Multiply Two Polynomials
- Strassen's Matrix Multiplication
- Karatsuba Algorithm for Fast Multiplication

Algorithms Content

Part 2: Graph Algorithms

Lecture 7, 8, 9, 10: Graph Algorithms

- Graph Traversal
- Topological Sort
- Connectivity
- Lowest Common Ancestor
- Single source shortest paths
- All pairs shortest paths
- Dijkstra
- Spanning trees
- Minimum Spanning Tree
- Matching
- Cycles
- Backtracking
- Maximum Flow
- Floyd Warshall
- Bellman Ford
- Kirchhoff Theorem
- Prim & Kruskal

Lecture 11, 12: Greedy Algorithms

- Standard Greedy Algorithms
- Greedy Algorithms in Graph
- Greedy Algorithms in Arrays
- Greedy Algorithms in Operating Systems

© Prepared by: Mohamed Ayman

Algorithms Content

Part 3: Mathematical Algorithms

Lecture 13, 14: Mathematical Algorithms

- Greatest Common Divisor (GCD)
- Latest Common Multiple (LCM)
- Prime Factorization and Divisors
- Chinese Remainder Theorem
- Sieve Algorithm
- Modular Arithmetic
- Euler Totient Function
- Number Theory
- nCr Computations
- Series

Lecture 15, 16: Geometric Algorithms

- Lines
- Polygon
- Circle
- Quickhull
- Triangle
- Rectangle
- Square
- Convex Hull
- Quadrilateral
- 3D Objects
- Plane Sweep
- Voronoi diagrams
- Delaunay triangulations

Lecture 17, 18, 19, 20: Computer Graphics Algorithms

- Line Generation Algorithm
- Circle Generation Algorithm
- Polygon Filling Algorithm
- Viewing & Clipping Algorithm
- 2D Transformation
- 3D Transformation
- Projection from 3D to 2D
- Computer Graphics Curves
- Computer Graphics Surfaces
- Visible Surface Detection
- Computer Graphics Fractals

© Prepared by: Mohamed Ayman

Algorithms Content

Part 4: Dynamic Programming

Lecture 21: Bitwise Algorithms

- Bit Manipulation
- Bitmasks Algorithm
- Bit Stuffing in Computer Networks
- Error Detection in Computer Networks

Lecture 22, 23, 24: Dynamic Programming

- Overlapping Sub-problems Property
- Optimal Sub-structure Property
- Tabulation vs Memoization
- Bitmasking & Dynamic Programming

Lecture 25, 26: Randomized Algorithms

- Randomized Quick Sort
- Monte Carlo Algorithms
- Las Vegas Algorithms
- Atlantic City Algorithms
- Computational Complexity

Algorithms Content

Part 5: String Algorithms

Lecture 27: String Algorithms

- Anagram
- Palindrome
- Binary String
- Subsequence
- Pattern Searching

Lecture 30, 31, 32: Pattern Searching Algorithms

- Naïve Pattern Searching
- KMP Algorithm
- Rabin-Karp Algorithm
- Finite Automata
- Boyer Moore Algorithm
- Z Algorithm
- Aho-Corasick Algorithm
- Kasai's Algorithm
- Anagram Substring Search
- Pattern Searching using a Trie of all Suffixes

Lecture 28, 29: String Compression Algorithms

- Lempel-Ziv Compression (LZ77 & LZ78)
- Lempel-Ziv-Markov Chain Algorithm (LZMA)
- Lempel-Ziv-Oberhumer (LZO)
- Lempel-Ziv-Storer-Szymanski (LZSS)
- Lempel-Ziv-Welch (LZW)
- Lempel-Ziv Finite State Entropy (LZFSE)
- Standard Huffman Coding Algorithm
- Modified Huffman Coding Algorithm
- Adaptive Huffman Coding Algorithm
- Arithmetic Coding (Float & Binary)

Hands-on Projects & Assignments & Practices

Algorithms Projects (6 Projects)

Project 1: Dictionary Sorting Simulator	(sorting application)
Project 2: Dictionary Searching Simulator	(searching application)
Project 3: Advanced Mathematical Calculator	(mathematical application)
Project 4: Advanced Geometric Simulator	(geometric application)
Project 5: Advanced Computer Graphics Generator	(computer graphics application)
Project 6: Advanced Computer Graphics Simulator	(computer graphics application)

Algorithms Assignments (8 Assignment)

- After each lecture we have an assignment (Implementing & Testing the Algorithms on each Lecture)

Algorithms Practices (30+ Practice Problems) on each Lecture.

© Prepared by: Mohamed Ayman

Lecture Agenda

- ✓ Section 1: Data Structures and Algorithms Features
- ✓ Section 2: Data Structures and Algorithms Content
- **Section 3: Practice on Online Judges**
- Section 4: Programming Competitions
- Section 5: Tutorials and References
- Section 6: Online Courses

Practice on Online Judges

CODEFORCES

codeforces.com

hackerearth

hackerearth.com

HackerRank

hackerrank.com

topcoder™

topcoder.com

AtCoder

atcoder.jp

UVa Online Judge

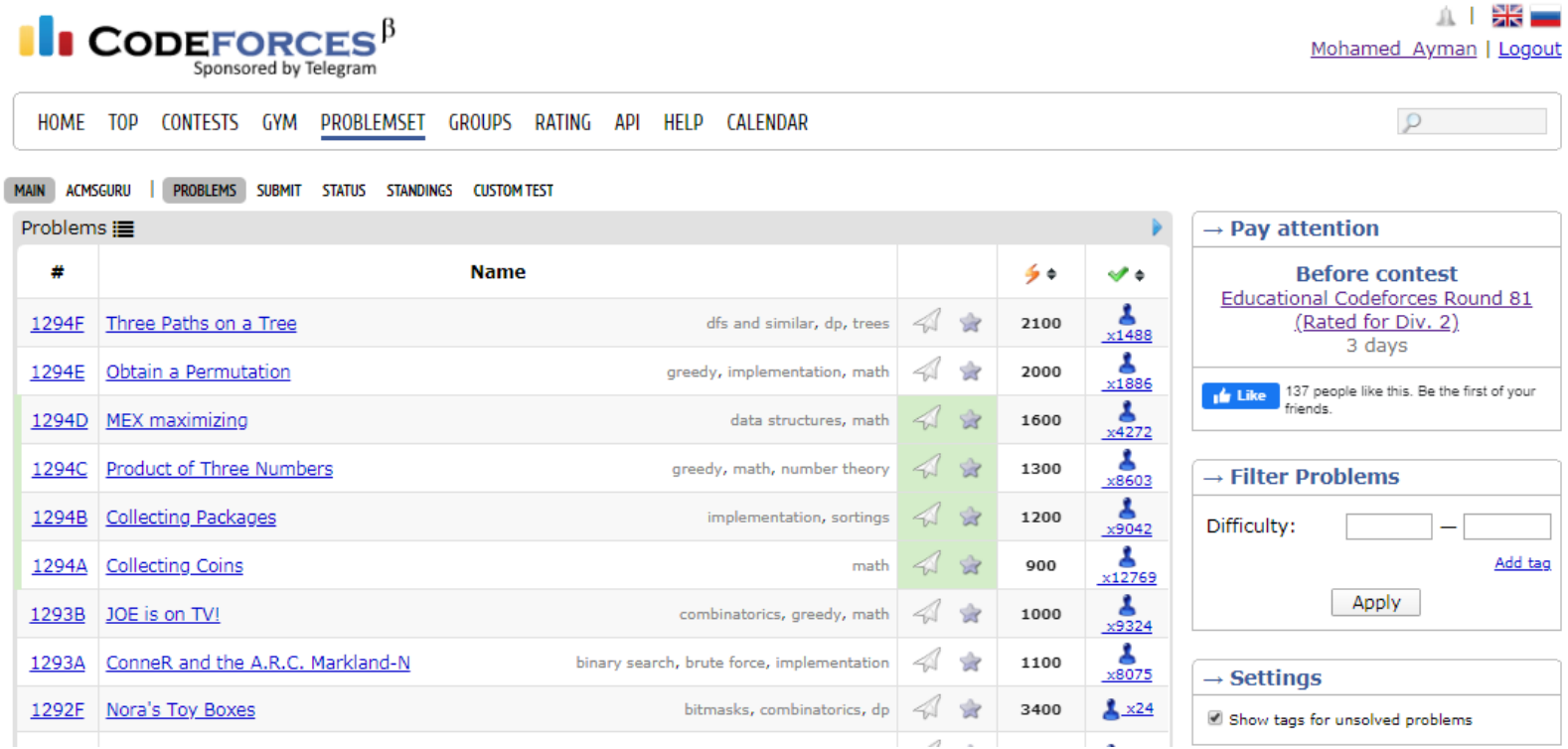
onlinejudge.org

© Prepared by: Mohamed Ayman

Codeforces Online Judge

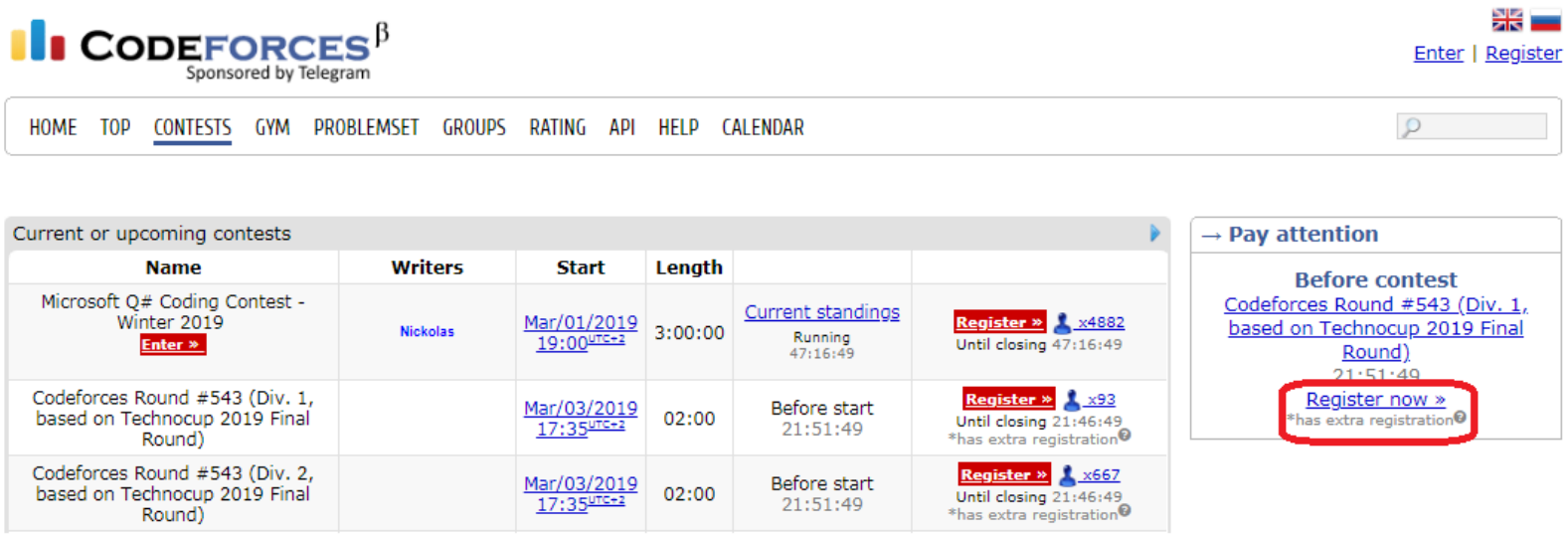
Codeforces is a website that hosts competitive programming contests. It is maintained by a group of competitive programmers from ITMO University led by Mikhail Mirzayanov.

Codeforces Online Judge



Register in New Contest

Register The Contest from Register now » link



Register in Previous Contest

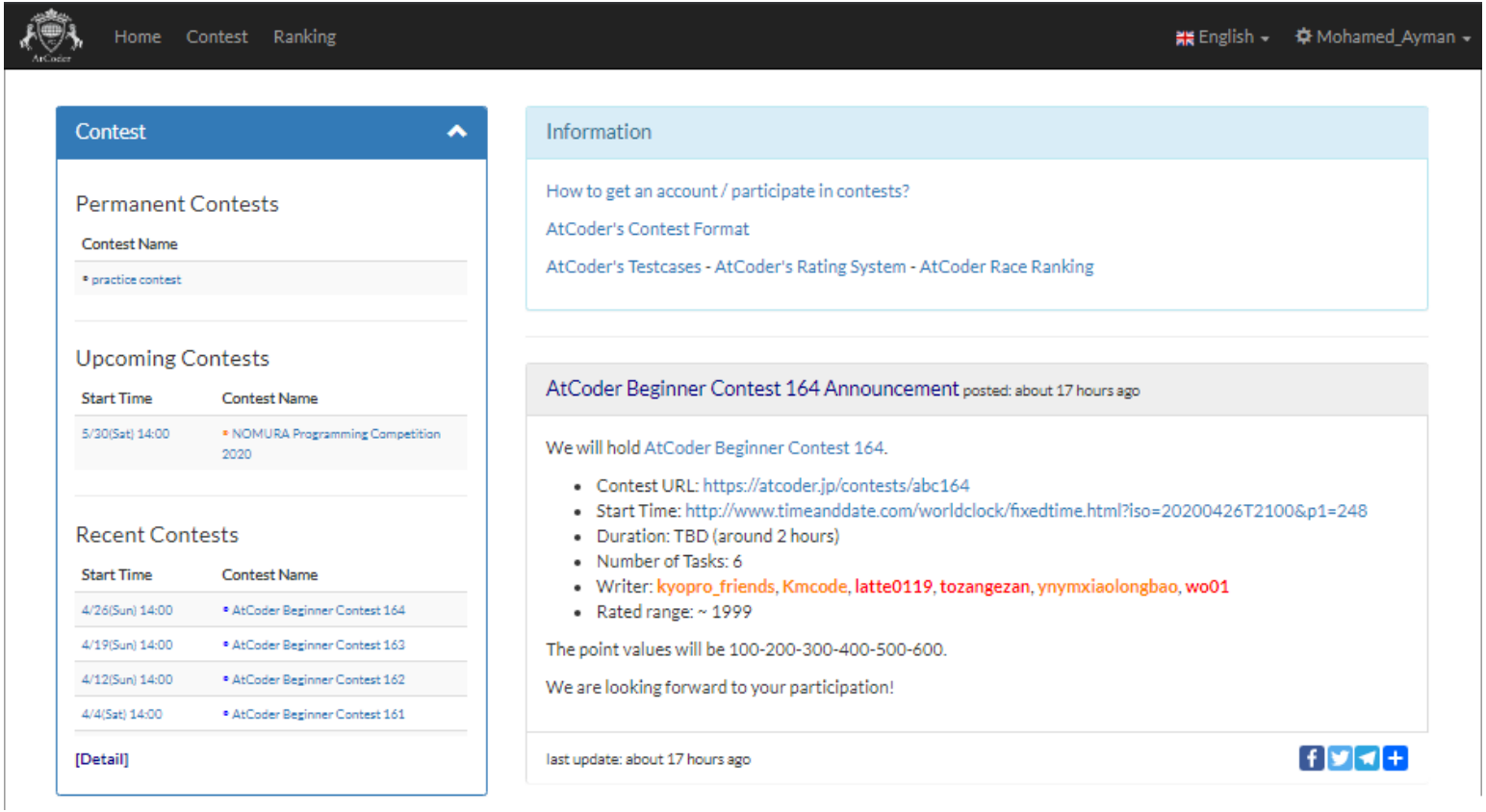
You Can Compete in Previous Contests

Contest history

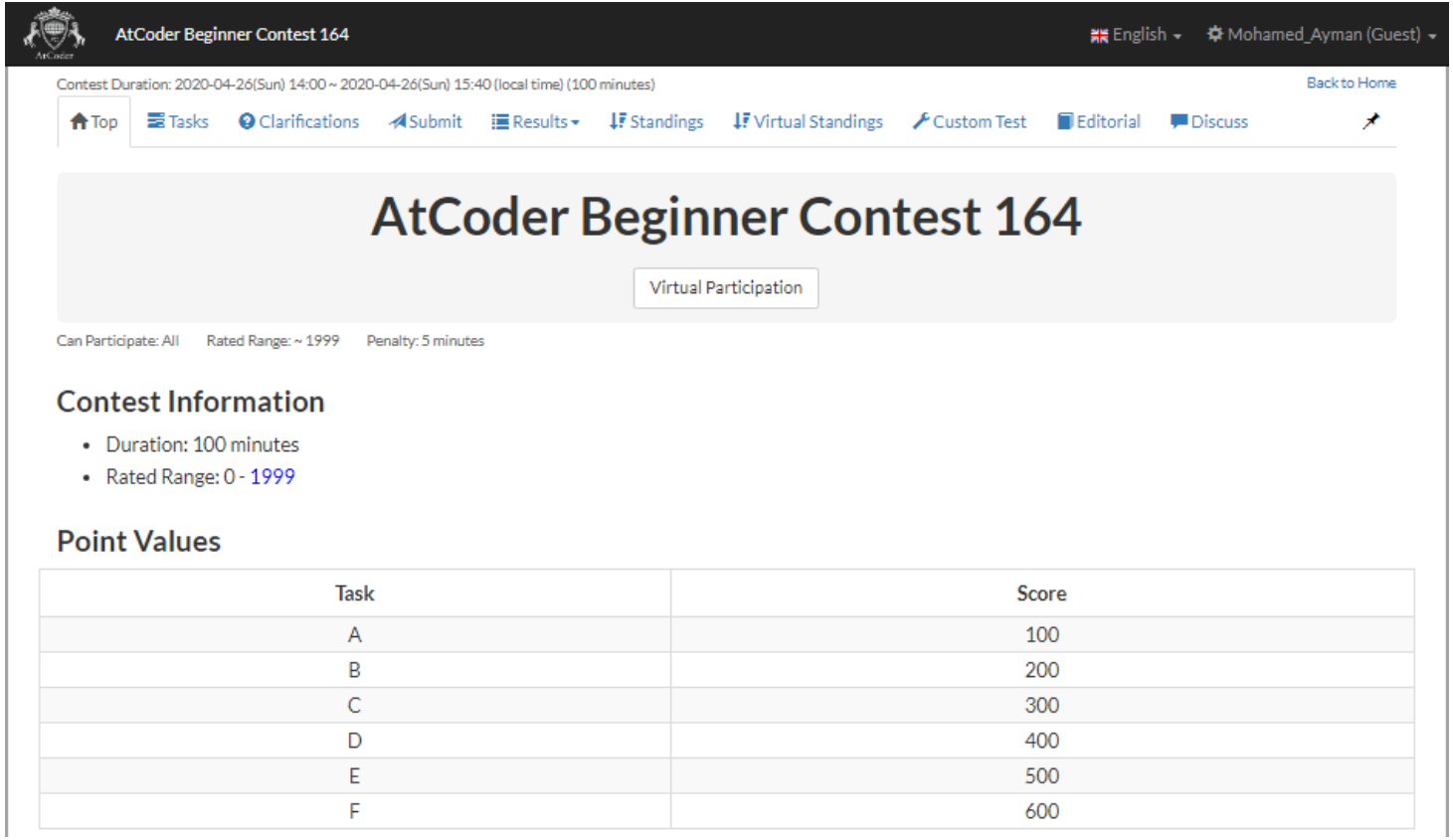
Past contests

Name	Writers	Start	Length		
Codeforces Round #542 [Alex Lopashev Thanks-Round] (Div. 1) Enter » Virtual participation »	top34051 zoomswk	Feb/24/2019 17:35 UTC+2	02:00	Final standings	x793
Codeforces Round #542 [Alex Lopashev Thanks-Round] (Div. 2) Enter » Virtual participation »	MikeMirzayanov top34051 zoomswk	Feb/24/2019 17:35 UTC+2	02:00	Final standings	x7221
Codeforces Round #541 (Div. 2) Enter » Virtual participation »	MikeMirzayanov Sehnsucht Sender V--gLaSsH0ldEr593--V VFeafanov _kun_ ch_egor grphil voidmax	Feb/23/2019 12:20 UTC+2	02:00	Final standings	x8568

AtCoder - Online Judge



AtCoder - Online Judge



AtCoder - Online Judge

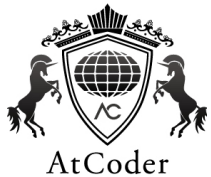
Contest Rules

This contest is full-feedback (solutions are judged during the contest).

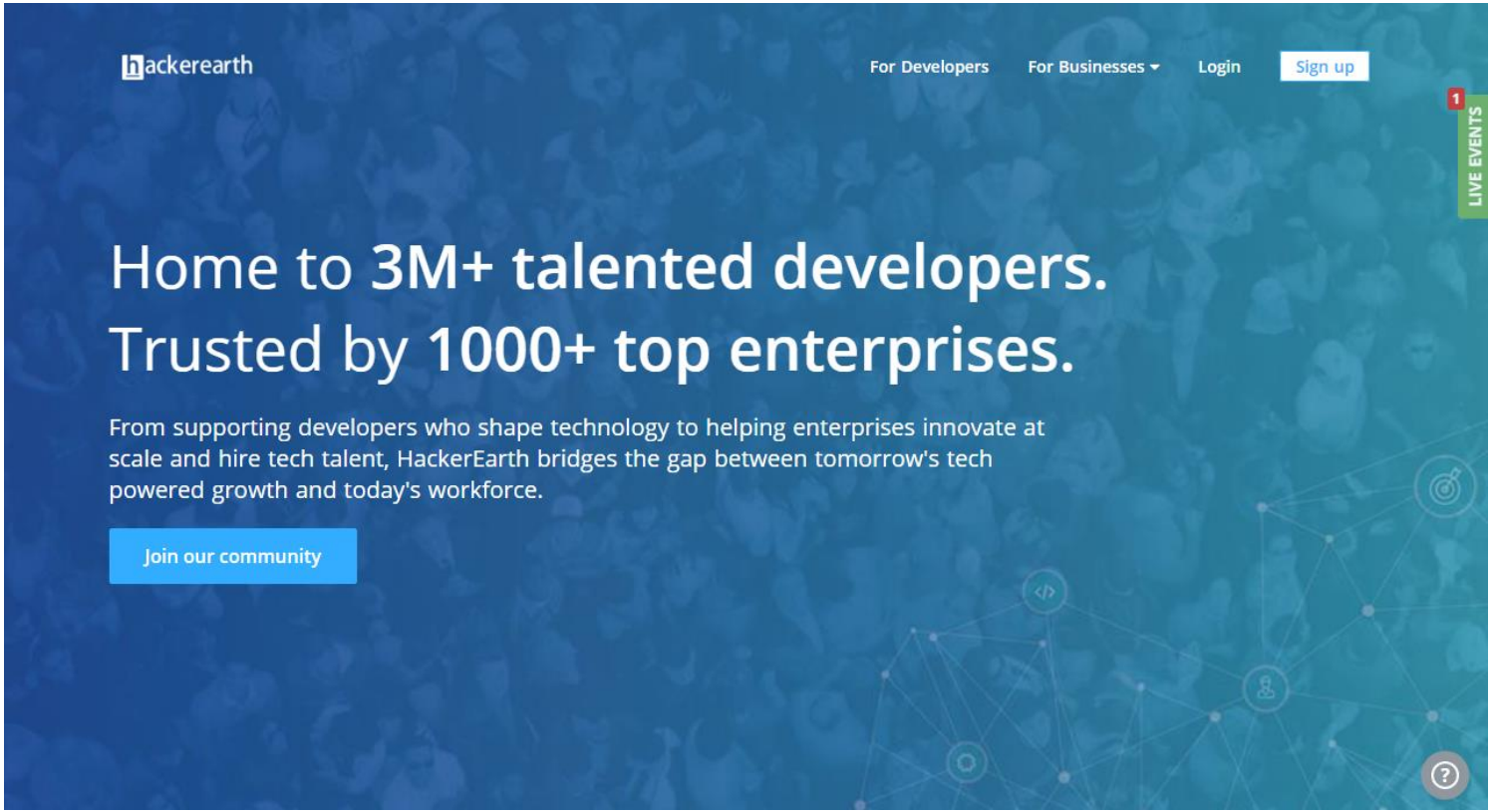
When you solve a problem, you get a score assigned to it. Competitors are ranked first by total scores, then by penalties. The penalties are computed as (the time you spend to get your current score) + (5 minutes) * (the number of incorrect attempts).

Useful Links

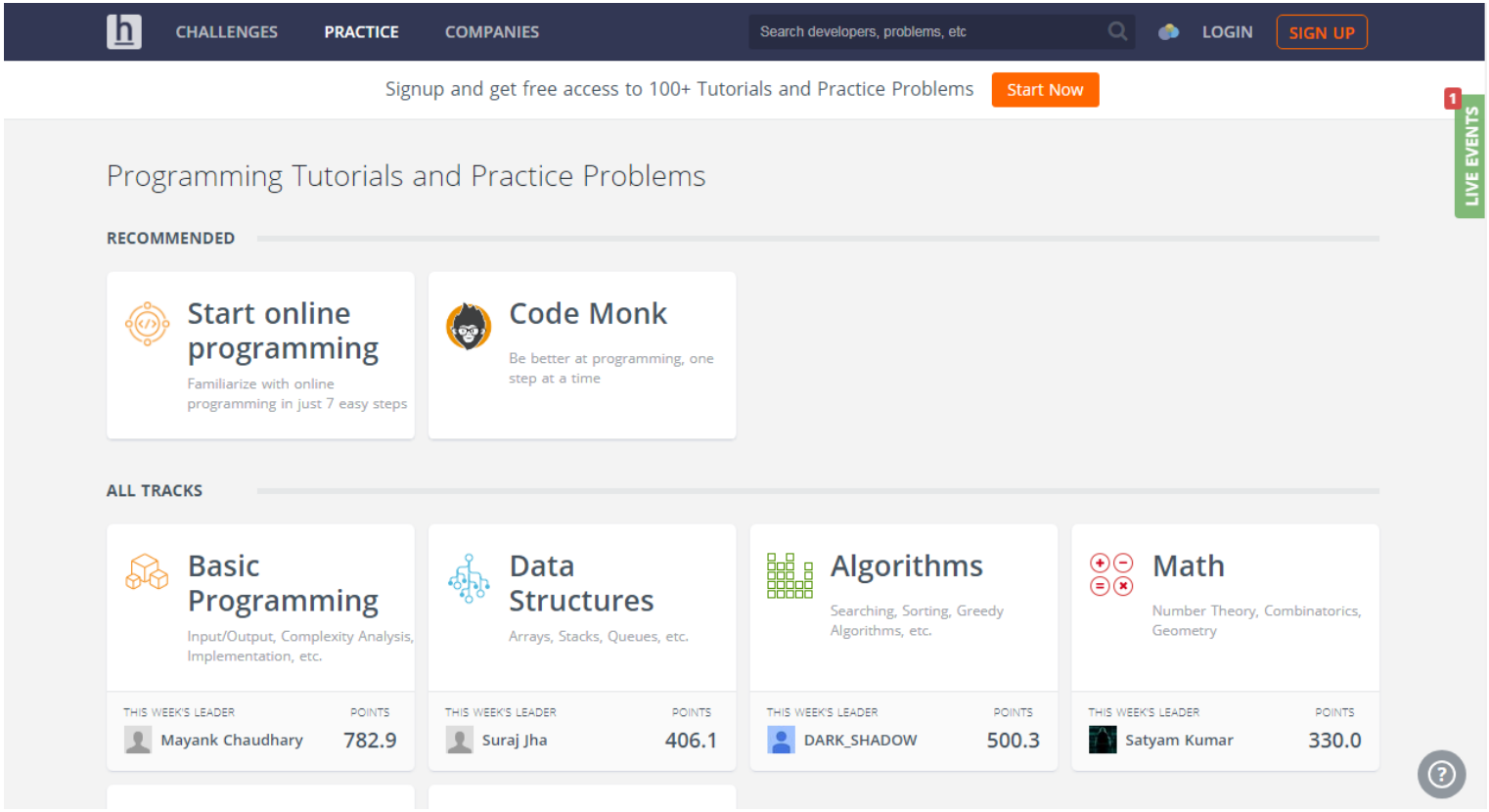
- AtCoder top page
- How to participate
- Practice contest



HackerEarth - Online Judge

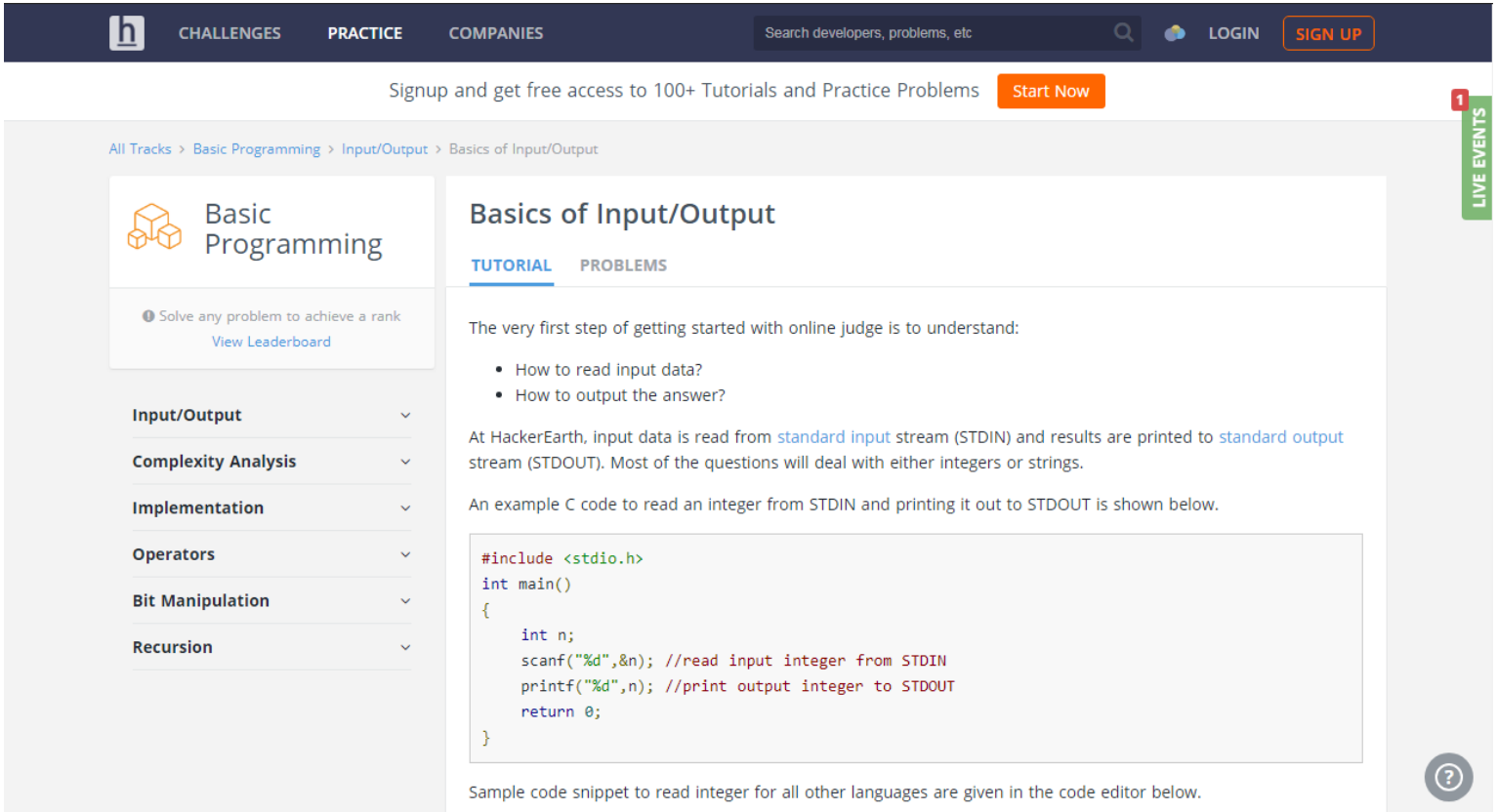


HackerEarth - Online Judge



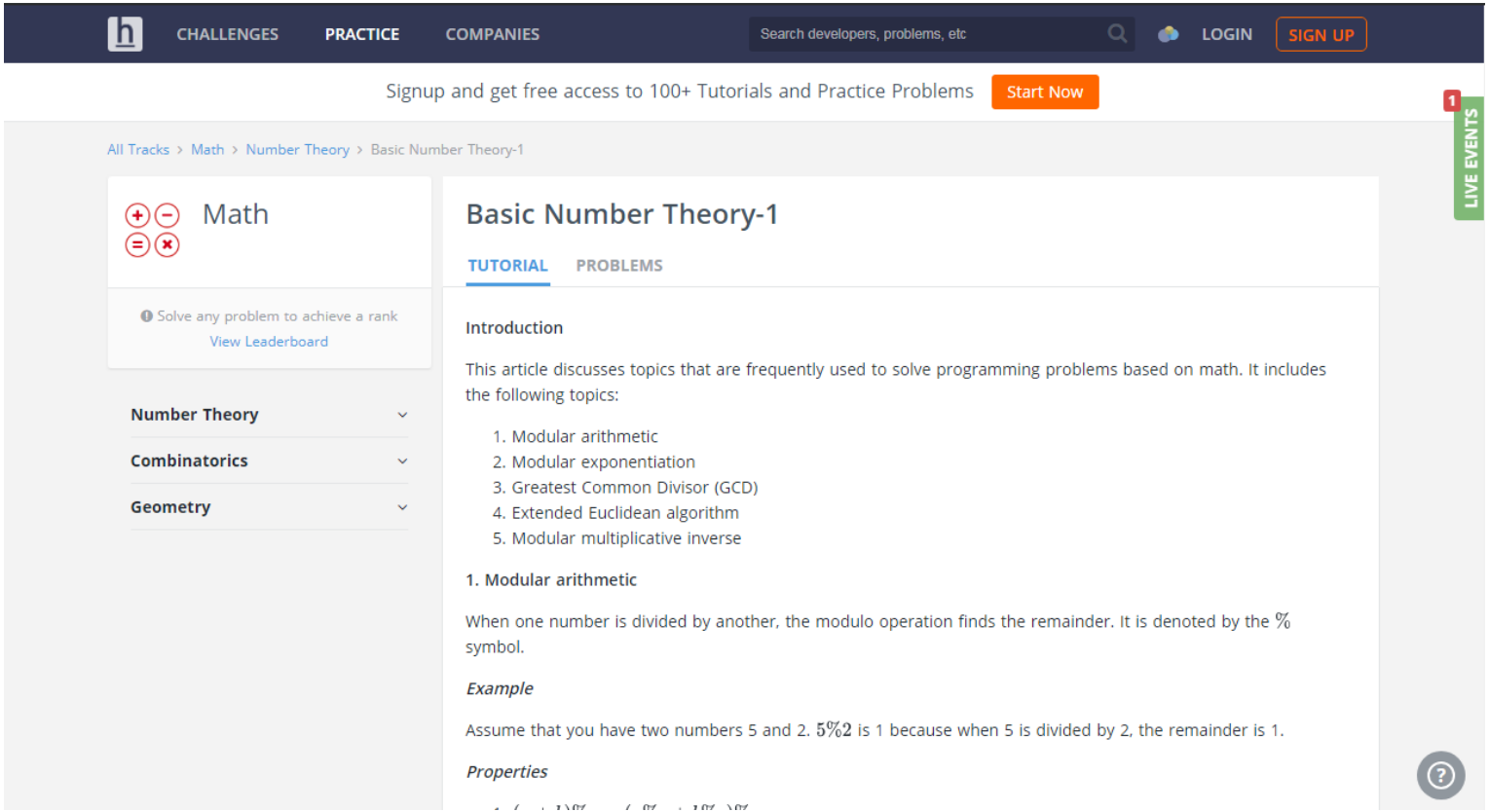
© Prepared by: Mohamed Ayman

HackerEarth - Online Judge

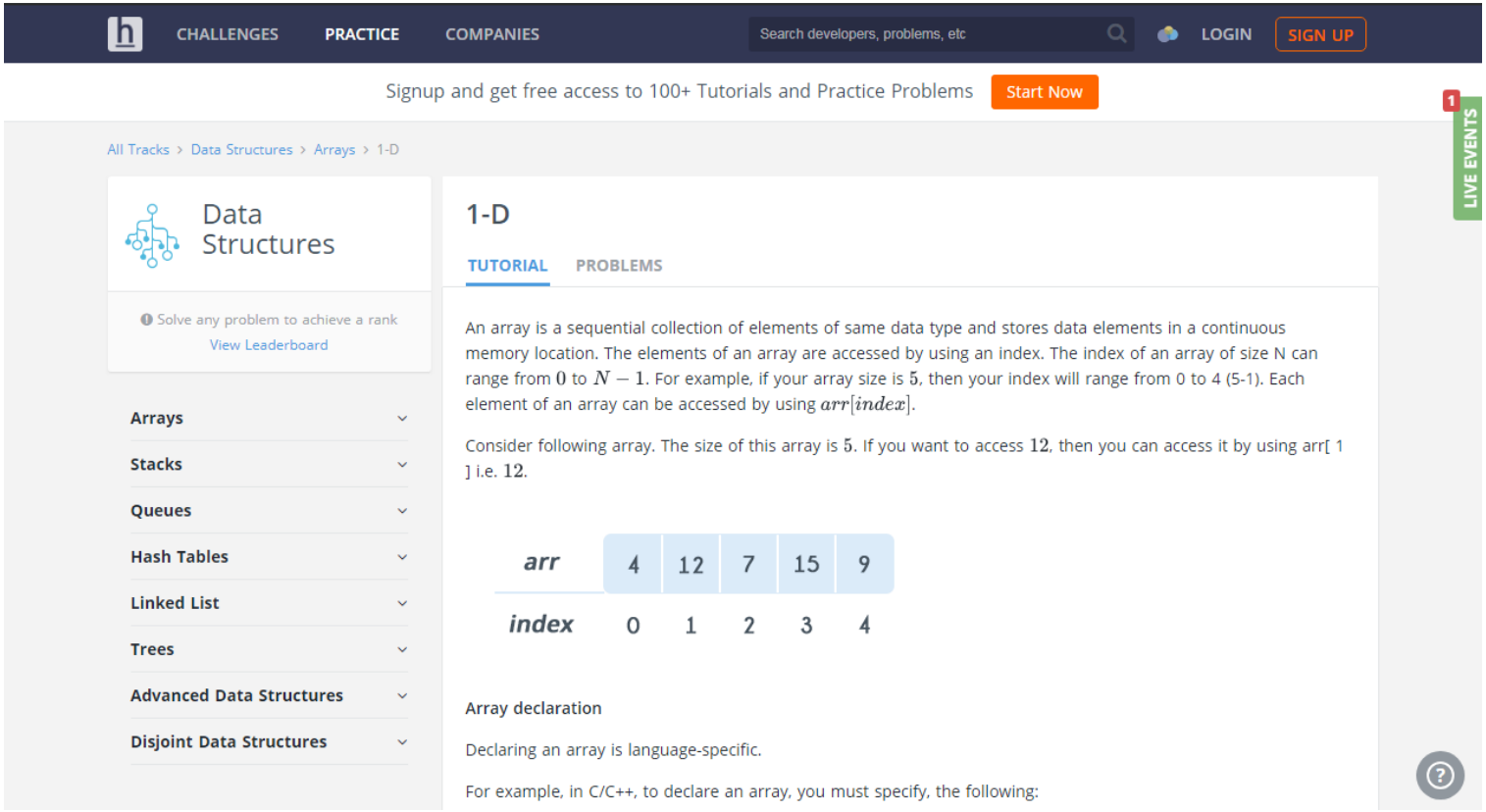


© Prepared by: Mohamed Ayman

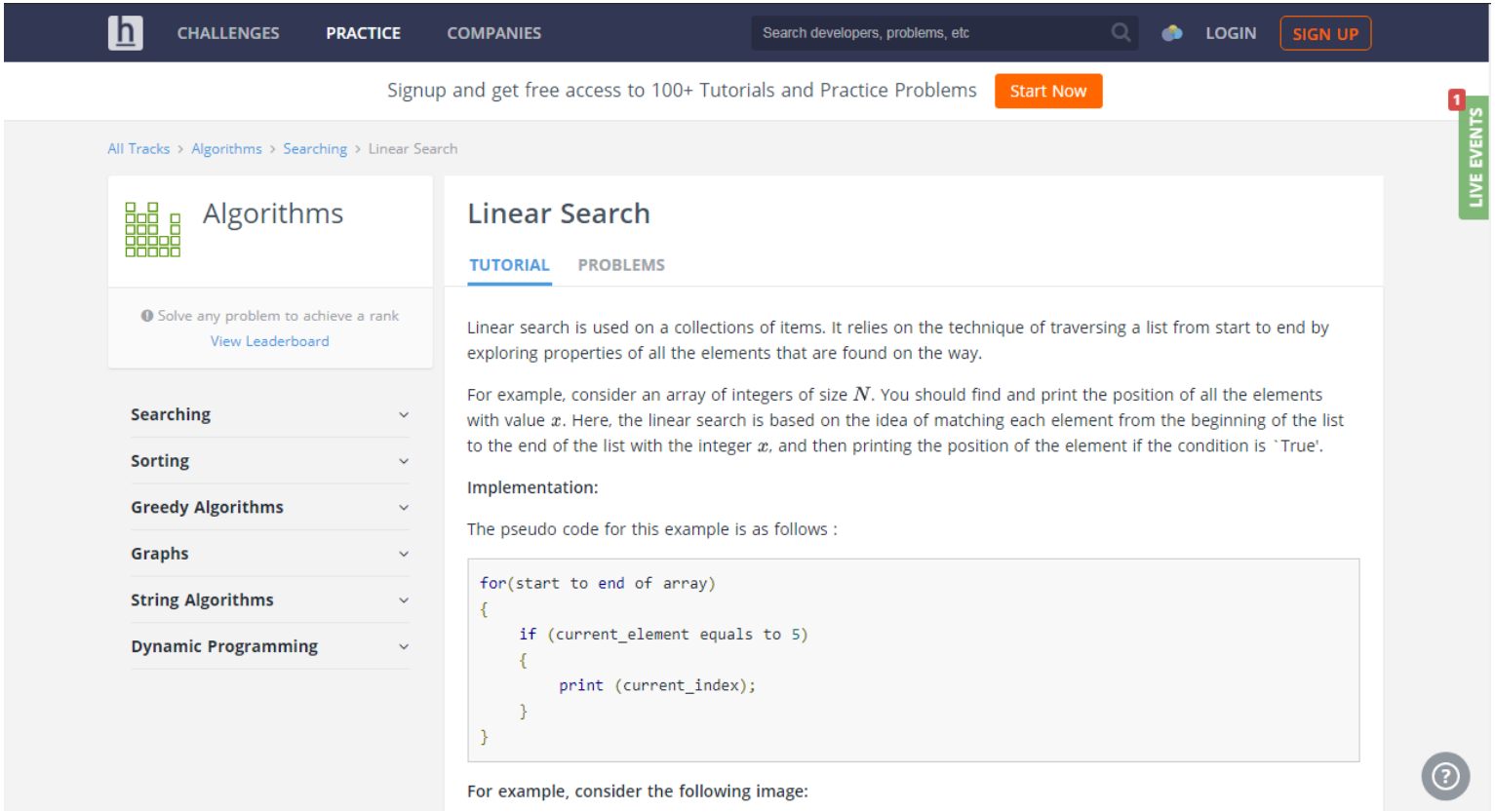
HackerEarth - Online Judge



HackerEarth - Online Judge

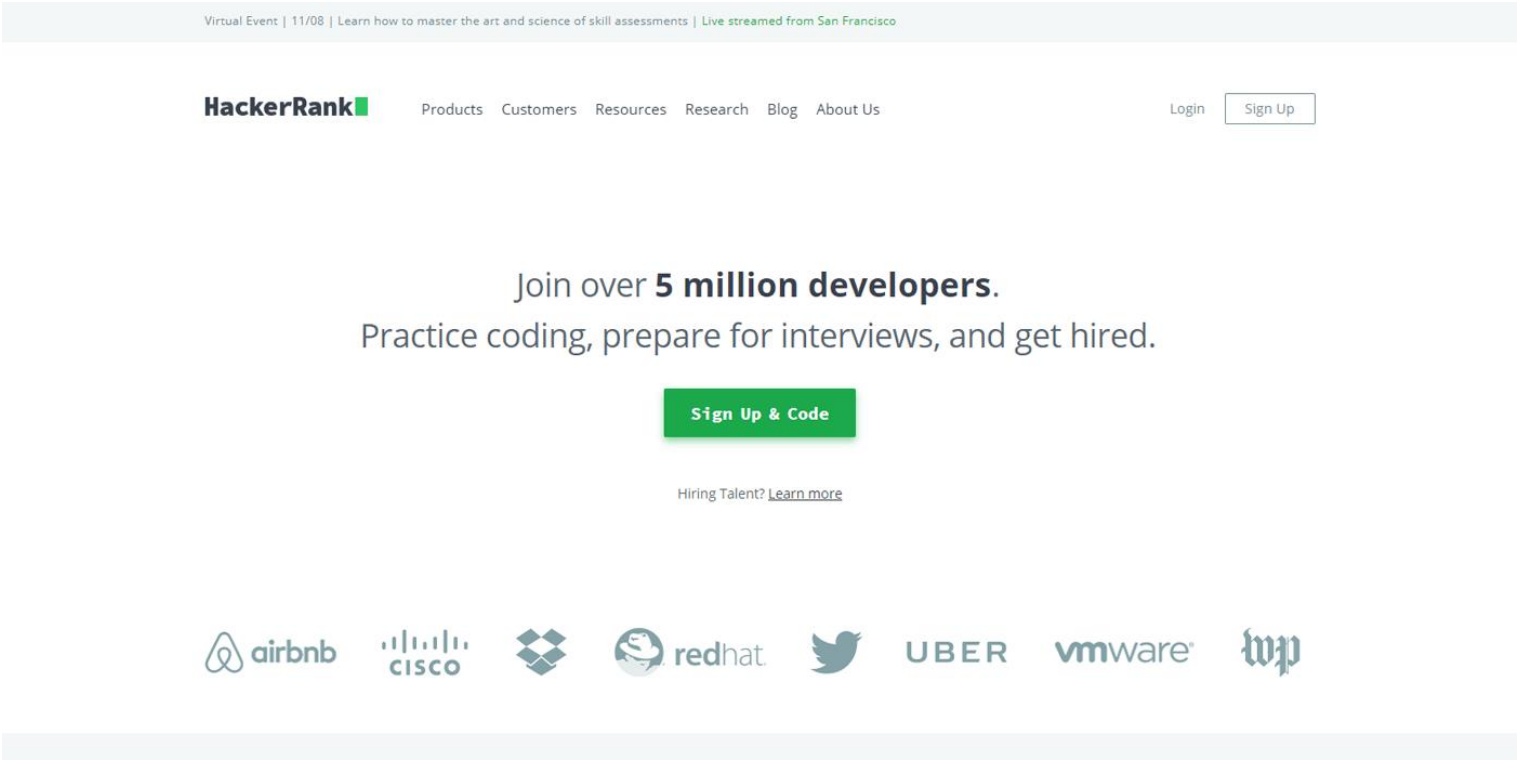


HackerEarth - Online Judge

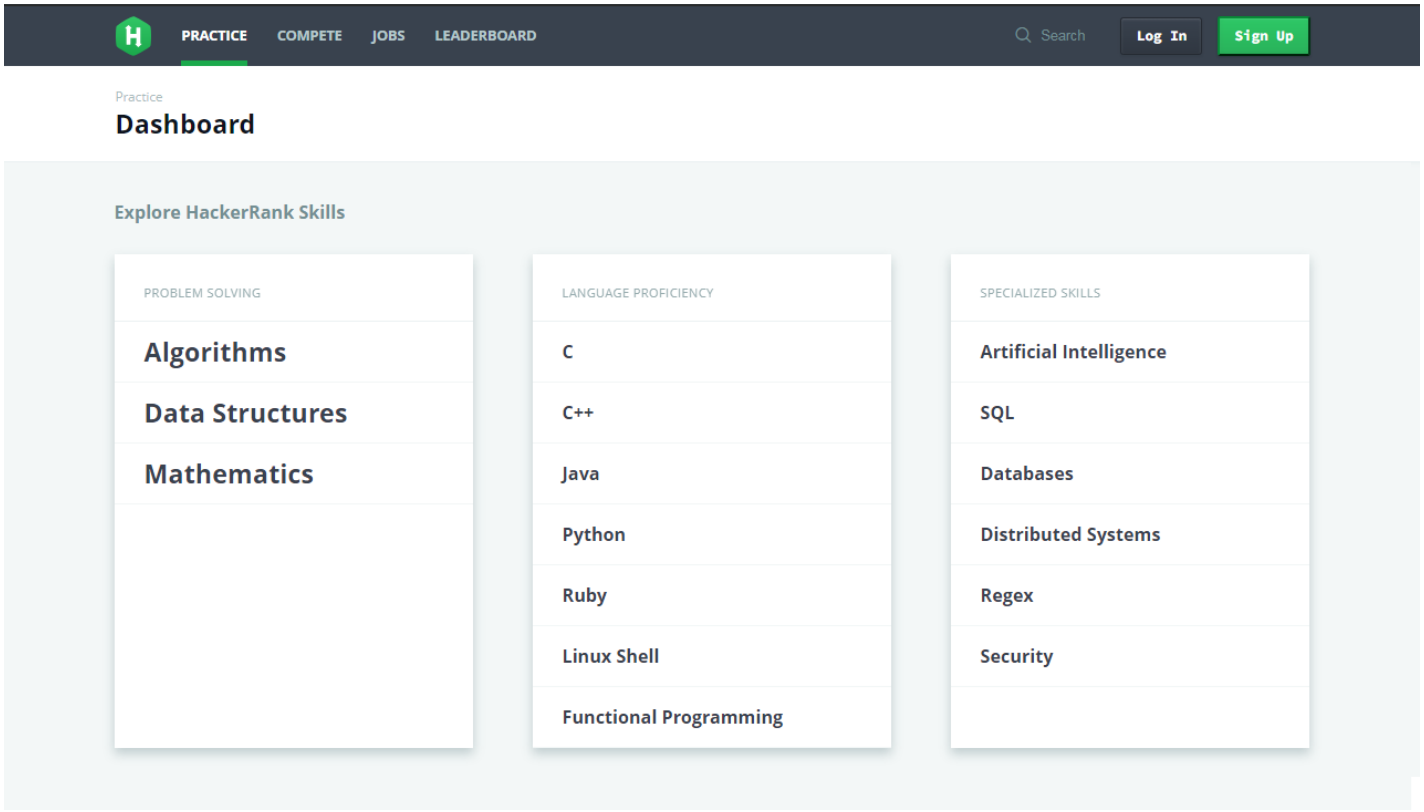


© Prepared by: Mohamed Ayman

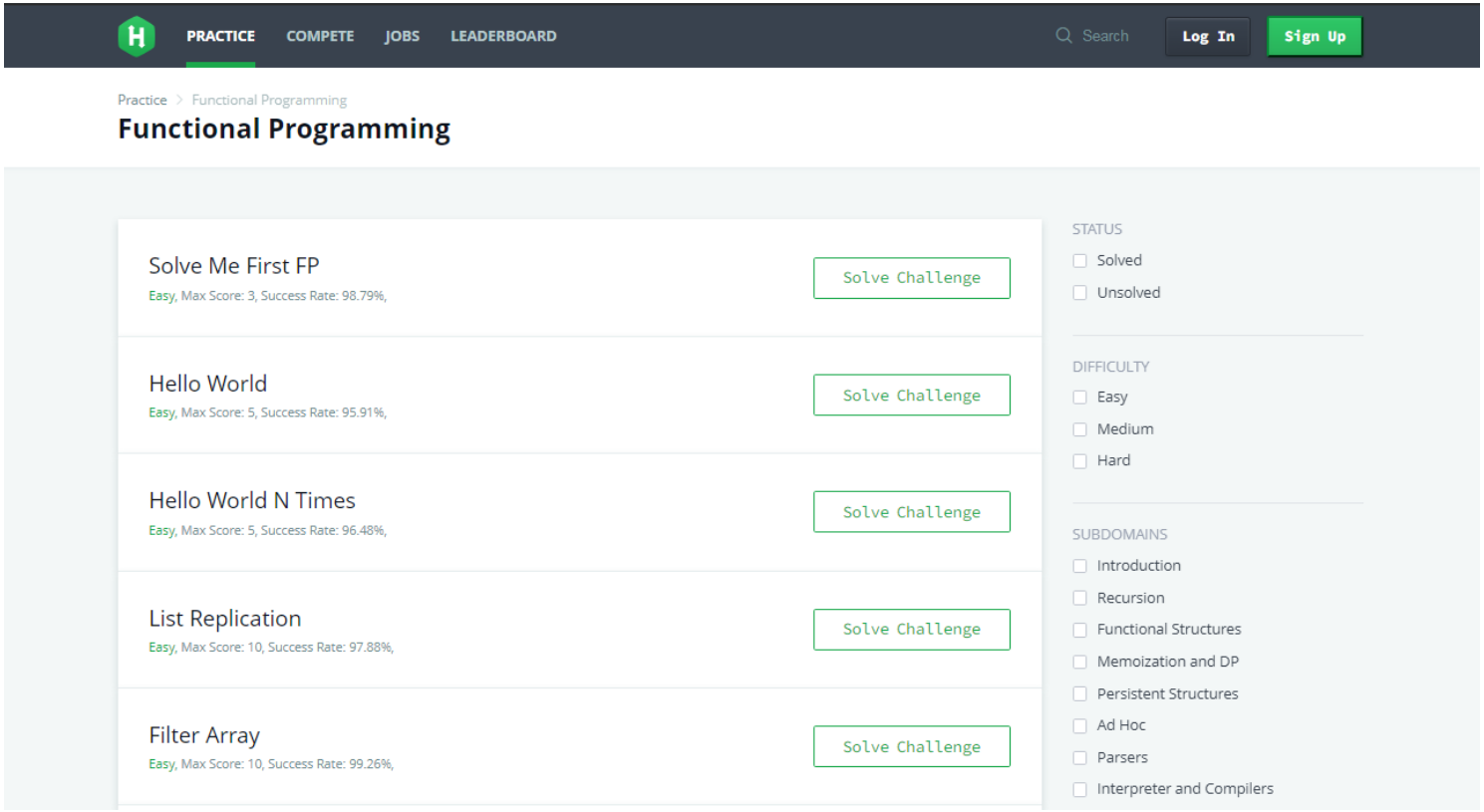
HackerRank - Online Judge



HackerRank - Online Judge

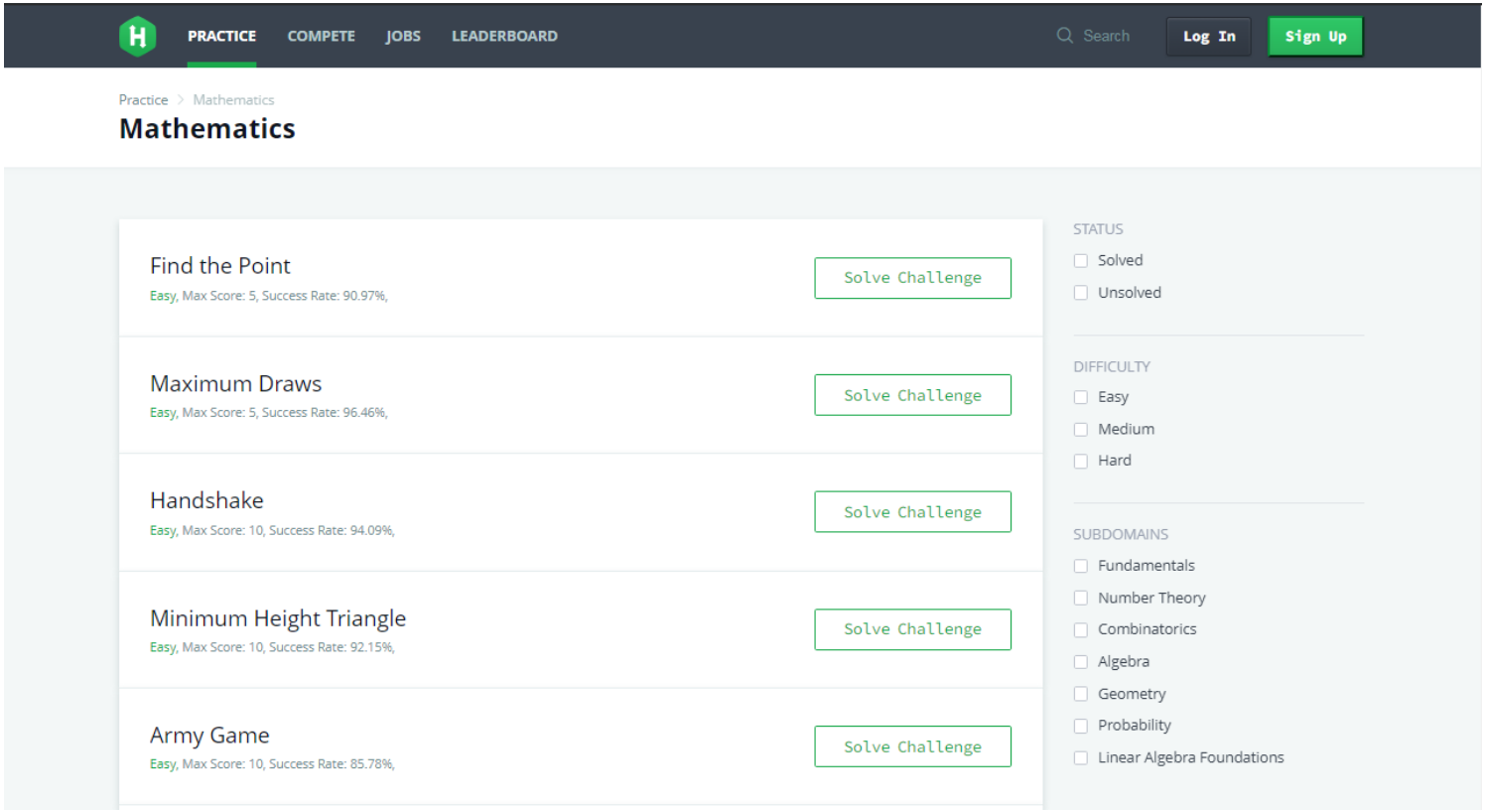


HackerRank - Online Judge



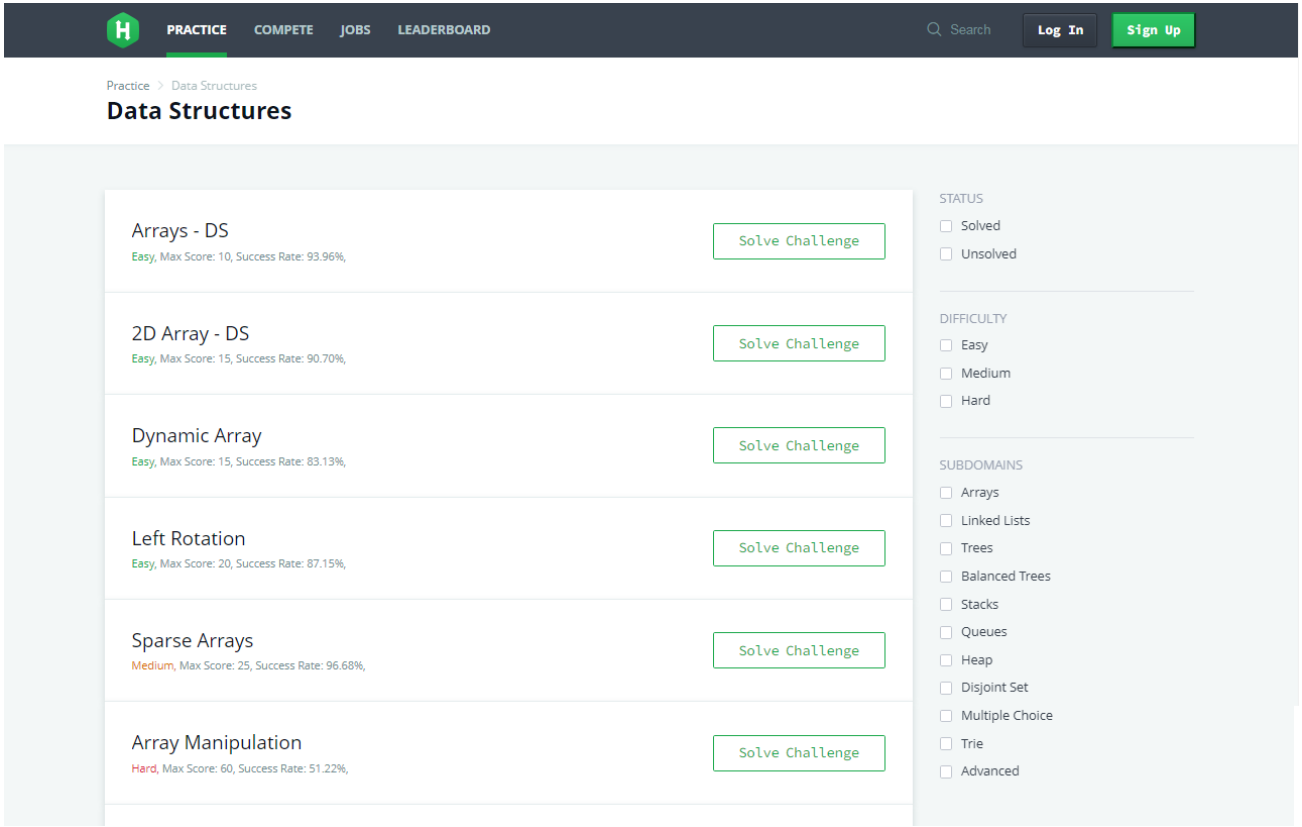
© Prepared by: Mohamed Ayman

HackerRank - Online Judge



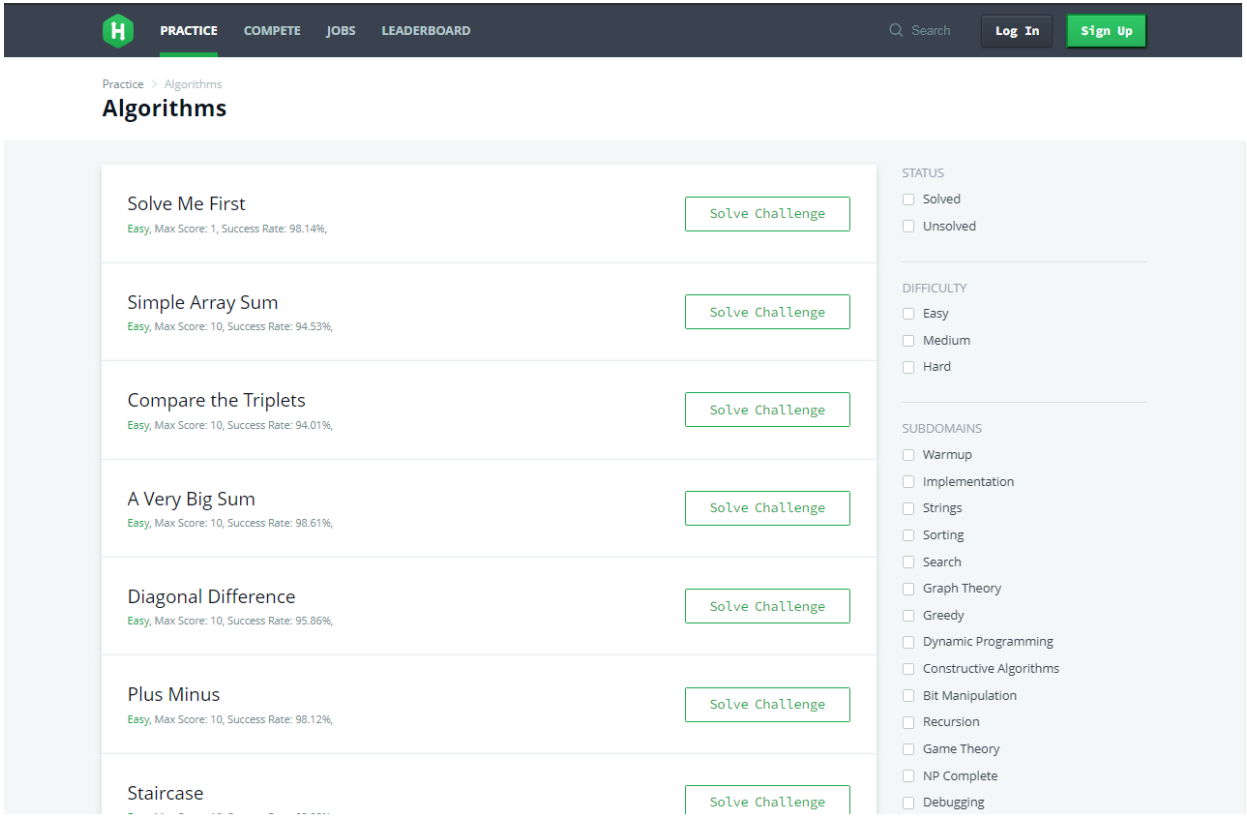
© Prepared by: Mohamed Ayman

HackerRank - Online Judge



© Prepared by: Mohamed Ayman

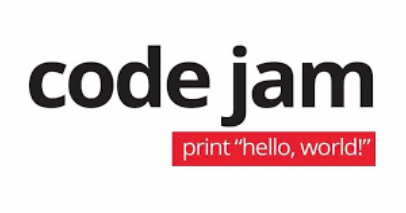
HackerRank - Online Judge

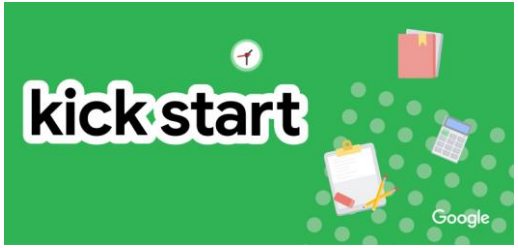


Lecture Agenda

- ✓ Section 1: Data Structures and Algorithms Features
- ✓ Section 2: Data Structures and Algorithms Content
- ✓ Section 3: Practice on Online Judges
- **Section 4: Programming Competitions**
- Section 5: Tutorials and References
- Section 6: Online Courses

Programming Competitions





Google Competitions

code jam

hash code

kick start

Google Competitions - Code Jam

Google Competitions - Code Jam

- Code Jam - Practice Session — March
- Code Jam - Qualification Round — March
- Code Jam - Round 1A — April
- Code Jam - Round 1B — April
- Code Jam - Round 1C — May
- Code Jam - Round 2 — May
- Code Jam - Round 3 — June
- Code Jam - World Finals — August

© Prepared by: Mohamed Ayman

Google Competitions - Kick Start

Google Competitions - Kick Start

- Kick Start - Round A — March
- Kick Start - Round B — April
- Kick Start - Round C — May
- Kick Start - Round D — July
- Kick Start - Round E — August
- Kick Start - Round F — September
- Kick Start - Round G — October
- Kick Start - Round H — November

© Prepared by: Mohamed Ayman

Google Competitions - Hash Code

Google Competitions - Hash Code

- Hash Code - Hub registration opens — November
- Hash Code - Individual registration opens — January
- Hash Code - Registration closes — February
- Hash Code - Online qualification round — February
- Hash Code - Results announced — March
- Hash Code - Final round — April

© Prepared by: Mohamed Ayman

Facebook Hacker Cup Competition

- Facebook Hacker Cup - Qualification round — June
- Facebook Hacker Cup - Round 1 — June
- Facebook Hacker Cup - Round 2 — July
- Facebook Hacker Cup - Round 3 — August
- Facebook Hacker Cup - Onsite Final — September

© Prepared by: Mohamed Ayman

ICPC - International College Programming Contest

- Qualification Round in Universities — September
- ECPC Egyptian College Programming Contest — October
- ACPC Arab College Programming Contest — January
- ICPC International College Programming Contest — May

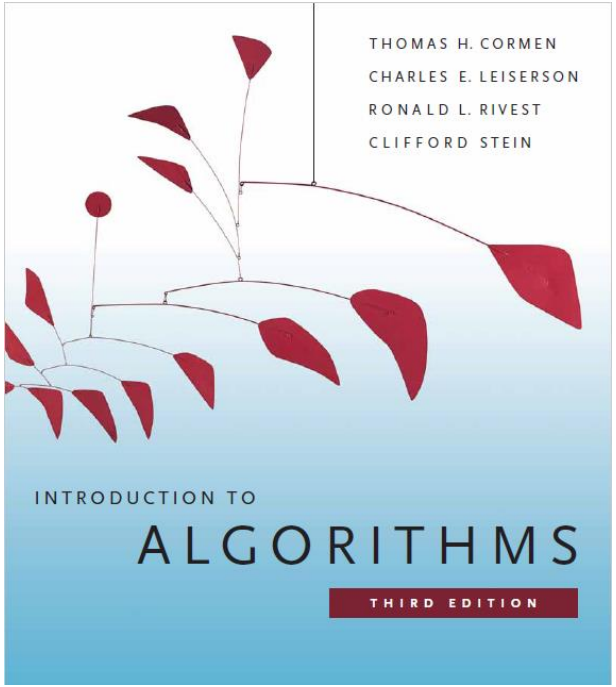
© Prepared by: Mohamed Ayman

Lecture Agenda

- ✓ Section 1: Data Structures and Algorithms Features
- ✓ Section 2: Data Structures and Algorithms Content
- ✓ Section 3: Practice on Online Judges
- ✓ Section 4: Programming Competitions
- **Section 5: Tutorials and References**
- Section 6: Online Courses

Introduction to Algorithms Thomas H. Cormen

- Foundations [005 - 145]
- Sorting and Order Statistics [145 - 220]
- Data Structures [230 - 350]
- Advanced Design and Analysis Techniques [355 - 460]
- Advanced Data Structures [480 - 575]
- Graph Algorithms [585 - 750]
- Selected Topics [770 - 1130]



Introduction to Algorithms Thomas H. Cormen

Data Structures and Algorithms Annotated Reference

- Introduction [01 - 10]
- Linked Lists [10 - 20]
- Binary Search Tree [20 - 30]
- Heap [30 - 40]
- Sets [40 - 50]
- Queues [50 - 55]
- AVL Tree [55 - 60]
- Sorting [60 - 70]
- Numeric [70 - 75]
- Searching [75 - 80]
- Strings [80 - 85]

Data Structures and Algorithms Annotated Reference

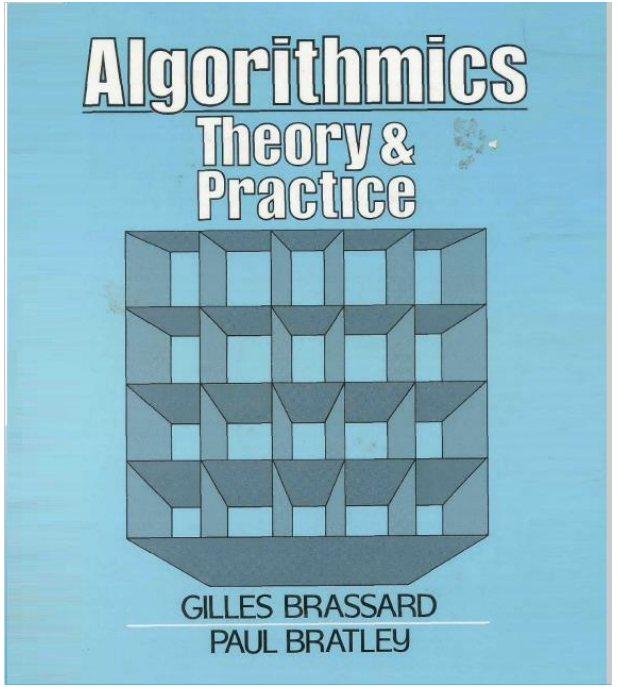
Competitive Programming 3 Steven Halim

- Introduction [001 - 030]
- Data Structures and Libraries [030 - 070]
- Problem Solving Paradigms [070 - 120]
- Graph [120 - 190]
- Mathematics [190 - 230]
- String Processing [230 - 270]
- (Computational) Geometry [270 - 300]
- More Advanced Topics [300 - 330]
- Rare Topics [330 - 390]

Competitive Programming 3 Steven Halim

Fundamental of Algorithmics Gilles Brassard

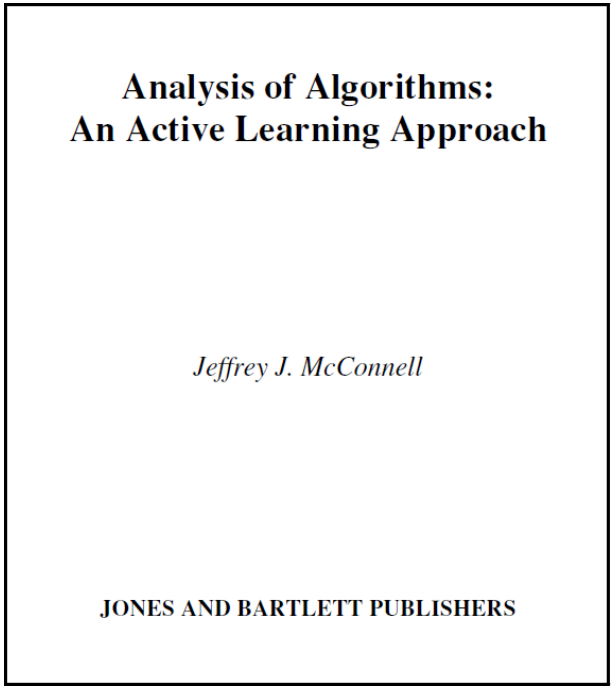
- Preliminaries [001 - 035]
- Analyzing the Efficiency of Algorithms [035 - 080]
- Greedy Algorithms [080 - 105]
- Divide and Conquer [105 - 140]
- Dynamic Programming [140 - 170]
- Exploring Graphs [170 - 205]
- Preconditioning and Pre-computation [205 - 225]
- Probabilistic Algorithms [225 - 275]
- Transformations of the Domain [275 - 290]
- Introduction to Complexity [290 - 335]



Fundamental of Algorithmics Gilles Brassard and Paul Bartley

Analysis of Algorithms An Active Learning Approach

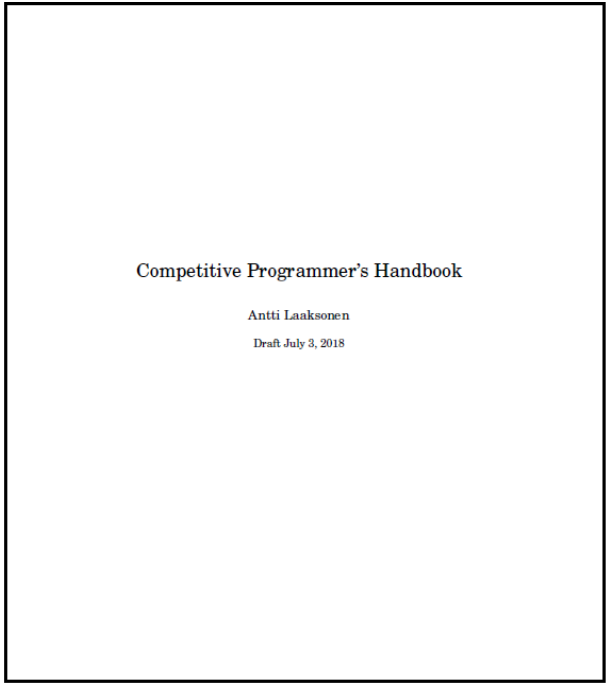
- Analysis Basics [001 - 040]
- Searching and Selection Algorithms [040 - 055]
- Sorting Algorithms [060 - 100]
- Numeric Algorithms [105 - 120]
- Matching Algorithms [120 - 140]
- Graph Algorithms [145 - 175]
- Parallel Algorithms [175 - 210]
- Nondeterministic Algorithms [210 - 230]
- Other Algorithmic Techniques [230 - 260]



Analysis of Algorithms An Active Learning Approach

Competitive Programmer's Handbook

- **Basic techniques** [001 - 105]
  - Time complexity
  - Complete search
  - Dynamic programming
  - Range queries
  - Sorting
  - Greedy algorithms
  - Amortized analysis
  - Bit manipulation
- **Graph algorithms** [105 - 195]
  - Graph traversal
  - Spanning trees
  - Tree queries
  - Shortest paths
  - Directed graphs
  - Paths and circuits
  - Tree algorithms
  - Strong connectivity
  - Flows and cuts
- **Advanced topics** [195 - 275]
  - Number theory
  - Game theory
  - Segment trees revisited
  - Combinatorics
  - String algorithms
  - Geometry
  - Matrices
  - Square root algorithms
  - Sweep line algorithms

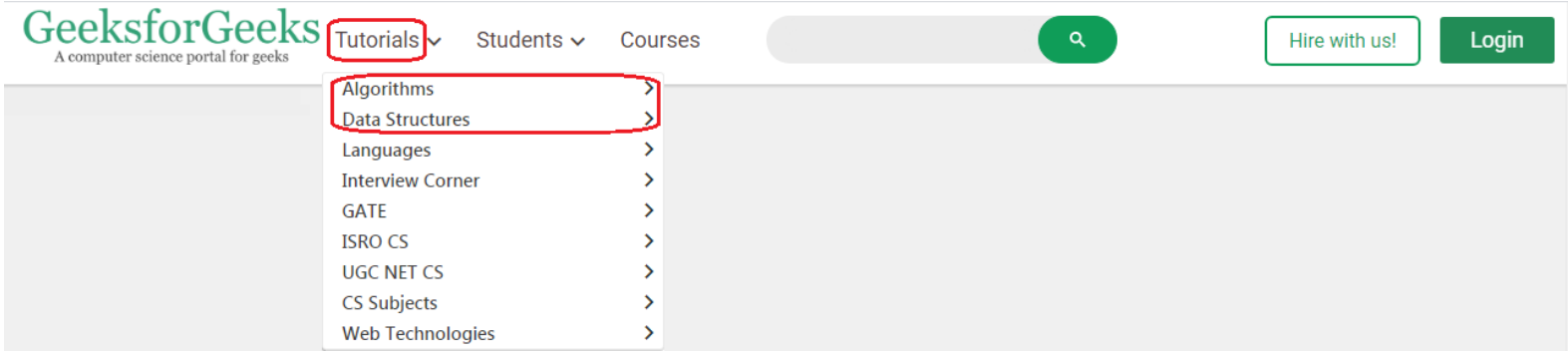


Competitive Programmer's Handbook

GeeksforGeeks Articles

GeeksforGeeks
A computer science portal for geeks

geeksforgeeks.org



Lecture Agenda

- ✓ Section 1: Data Structures and Algorithms Features
- ✓ Section 2: Data Structures and Algorithms Content
- ✓ Section 3: Practice on Online Judges
- ✓ Section 4: Programming Competitions
- ✓ Section 5: Tutorials and References

Section 6: Online Courses

Data Structures & Algorithms Specializations

- Accelerated Computer Science Fundamentals Specialization (3 Courses)

by University of Illinois at Urbana-Champaign coursera.org/specializations/cs-fundamentals

Course : Object-Oriented Data Structures in C++

Week 1: Orientation; Writing a C++ Program
Week 2: Understanding the C++ Memory Model
Week 3: Developing C++ Classes
Week 4: Engineering C++ Software Solutions

Course : Ordered Data Structures

Week 1: Orientation; Linear Structures
Week 2: Introduction to Tree Structures
Week 3: Advanced Tree Structures
Week 4: Heap Structures

Course : Unordered Data Structures

Week 1: Orientation; Hashing
Week 2: Disjoint Sets
Week 3: Graph Data Structures
Week 4: Graph Algorithms

© Prepared by: Mohamed Ayman

Data Structures & Algorithms Specializations

- Algorithms Specialization (4 Courses) by Stanford University coursera.org/specializations/algorithms

Course : Divide and Conquer, Sorting and Searching, and Randomized Algorithms

Week 1: Introduction, big-oh notation and asymptotic analysis
Week 2: Divide and conquer basics, the master method for analyzing divide and conquer algorithms
Week 3: The QuickSort algorithm and its analysis, probability review
Week 4: Linear-time selection, graphs, cuts, and the contraction algorithm

Course : Graph Search, Shortest Paths, and Data Structures

Week 1: Breadth-first and depth-first search, computing strong components, applications
Week 2: Dijkstra's shortest-path algorithm
Week 3: Heaps, balanced binary search trees
Week 4: Hashing, bloom filters

Data Structures & Algorithms Specializations

- Algorithms Specialization (4 Courses) by Stanford University coursera.org/specializations/algorithms

Course : Greedy Algorithms, Minimum Spanning Trees, and Dynamic Programming

Week 1: Two motivating applications, selected review, introduction to greedy algorithms, a scheduling application, Prim's MST algorithm
Week 2: Kruskal's MST algorithm and applications to clustering, advanced union-find
Week 3: Huffman codes, introduction to dynamic programming
Week 4: Advanced dynamic programming: the knapsack problem, sequence alignment, and optimal binary search trees

Course : Simulation, Algorithm Analysis, and Pointers

Week 1: The Bellman-Ford algorithm, all-pairs shortest paths
Week 2: NP-complete problems and exact algorithms for them
Week 3: Approximation algorithms for NP-complete problems
Week 4: Local search algorithms for NP-complete problems, the wider world of algorithms

Data Structures & Algorithms Specializations

- Data Structures and Algorithms Specialization (6 Courses)

by University of California San Diego & National Research University Higher School of Economics

coursera.org/specializations/data-structures-algorithms

Course : Algorithmic Toolbox

Week 1: Programming Challenges
Week 2: Algorithmic Warm-up
Week 3: Greedy Algorithms
Week 4: Divide-and-Conquer
Week 5: Dynamic Programming 1
Week 6: Dynamic Programming 2

Course : Data Structures

Week 1: Basic Data Structures
Week 2: Dynamic Arrays and Amortized Analysis
Week 3: Priority Queues and Disjoint Sets
Week 4: Hash Tables
Week 5: Binary Search Trees
Week 6: Binary Search Trees 2

Data Structures & Algorithms Specializations

- Data Structures and Algorithms Specialization (6 Courses)

by University of California San Diego & National Research University Higher School of Economics

coursera.org/specializations/data-structures-algorithms

Course : Algorithms on Graphs

Week 1: Decomposition of Graphs 1
Week 2: Decomposition of Graphs 2
Week 3: Paths in Graphs 1
Week 4: Paths in Graphs 2
Week 5: Minimum Spanning Trees
Week 6: Advanced Shortest Paths Project

Course : Algorithms on Strings

Week 1: Suffix Trees
Week 2: Burrows-Wheeler Transform and Suffix Arrays
Week 3: Knuth–Morris–Pratt Algorithm
Week 4: Constructing Suffix Arrays and Suffix Trees

Data Structures & Algorithms Specializations

- Data Structures and Algorithms Specialization (6 Courses)

by University of California San Diego & National Research University Higher School of Economics

coursera.org/specializations/data-structures-algorithms

Course : Advanced Algorithms and Complexity

- Week 1: Flows in Networks
- Week 2: Linear Programming
- Week 3: NP-complete Problems
- Week 4: Coping with NP-completeness
- Week 5: Streaming Algorithms

Course : Genome Assembly Programming Challenge

- Week 1: The 2011 European E. coli Outbreak
- Week 2: Assembling Genomes Using de Bruijn Graphs
- Week 3: Genome Assembly Faces Real Sequencing Data

Data Structures & Algorithms Specializations

- Algorithms, Part I by Princeton University

coursera.org/learn/algorithms-part1

Course : Algorithms, Part I

Week 1: Course Introduction
Union-Find
Analysis of Algorithms
Week 2: Stacks and Queues
Elementary Sorts
Week 3: Merge sort
Quick sort
Week 4: Priority Queues
Elementary Symbol Tables
Week 5: Balanced Search Trees
Geometric Applications of BSTs
Week 6: Hash Tables
Symbol Table Applications

- Algorithms, Part II by Princeton University

coursera.org/learn/algorithms-part2

Course : Algorithms, Part II

Week 1: Introduction
Undirected Graphs
Directed Graphs
Week 2: Minimum Spanning Trees
Shortest Paths
Week 3: Maximum Flow and Minimum Cut
Radix Sorts
Week 4: Tries
Substring Search
Week 5: Regular Expressions
Data Compression
Week 6: Reductions
Linear Programming (optional)
Intractability

Data Structures & Algorithms Specializations

- Geometric Algorithms by EIT Digital coursera.org/learn/geometric-algorithms

Course : Geometric Algorithms

Week 1: Plane Sweep Algorithms
Week 2: Voronoi diagrams and Delaunay triangulations
Week 3: Orthogonal range searching

- Approximation Algorithms by EIT Digital coursera.org/learn/approximation-algorithms

Course : Approximation Algorithms

Week 1: Point inclusion in a polygon
Week 2: Convex hulls
Week 3: Intersections
Week 4: Polygon triangulation
Week 5: Orthogonal range search

Data Structures & Algorithms Specializations

- Analysis of Algorithms by Princeton University

coursera.org/learn/analysis-of-algorithms

Course : Analysis of Algorithms

Week 1: Analysis of Algorithms
Week 2: Recurrences
Week 3: Generating Functions
Week 4: Asymptotics
Week 5: Analytic Combinatorics
Week 6: Trees
Week 7: Permutations
Week 8: Strings and Tries
Week 9: Words and Mappings

- Computational Geometry
by Saint Petersburg State University

coursera.org/learn/computational-geometry

Course : Computational Geometry

Week 1: Point inclusion in a polygon
Week 2: Convex hulls
Week 3: Intersections
Week 4: Polygon triangulation
Week 5: Orthogonal range search

Data Structures and Algorithms Playlists

- Playlist: Arrays | Data Structures & Algorithms [100 videos] [5 min] Channel: GeeksforGeeks
  youtube.com/playlist?list=PLqM7alHXFySEQDk2MDfbwEdjd2svVJH9p
- Playlist: Linked List | Data Structures & Algorithms [60 videos] [5 min] Channel: GeeksforGeeks
  youtube.com/playlist?list=PLqM7alHXFySH41ZxzrPNj2pAYPOI8ITe7
- Playlist: Stack | Data Structures & Algorithms [20 videos] [5 min] Channel: GeeksforGeeks
  youtube.com/playlist?list=PLqM7alHXFySF7Lap-wi5qlaD8OEBx9RMV
- Playlist: Queue | Data Structures & Algorithms [10 videos] [5 min] Channel: GeeksforGeeks
  youtube.com/playlist?list=PLqM7alHXFySG6wgjVeEat ouTIi0IBQ6D
- Playlist: Graph | Data Structures & Algorithms [30 videos] [5 min] Channel: GeeksforGeeks
  youtube.com/playlist?list=PLqM7alHXFySEaZgcg7uRYJFBnYMLti-nh
- Playlist: Trees | Data Structures & Algorithms [200 videos] [5 min] Channel: GeeksforGeeks
  youtube.com/playlist?list=PLqM7alHXFySHCXD7r1J0ky9Zg_GBB1dbk
- Playlist: Matrix | Data Structures & Algorithms [10 videos] [10 min] Channel: GeeksforGeeks
  youtube.com/playlist?list=PLqM7alHXFySGNyLyr8A2CBEBIbUIEC38f
- Playlist: Hashing | Data Structures & Algorithms [10 videos] [5 min] Channel: GeeksforGeeks
  youtube.com/playlist?list=PLqM7alHXFySGwXaessYMemAnITqIZdZVE

Data Structures and Algorithms Playlists

- Playlist: Data Structures [90 videos] [10 min] Channel: RobEdwardsSDSU
  youtube.com/playlist?list=PLpPXw4zFa0uKKhaSz87IowJn0Tzh9tiBk
- Playlist: Data Structure(ETCS - 209) - IP University Syllabus [60 videos] [10 min] Channel: Easy Engineering Classes
  youtube.com/playlist?list=PLV8vIYTIdSna11Vc54-abg33JtVZiiMfg
- Playlist: Data Structures and Algorithms [70 videos] [10 min] Channel: Gate Instructors
  youtube.com/playlist?list=PLXVjII7-2kRkrIwIVmSTF236m3z9sRCr8
- Playlist: Data Structures [40 videos] [15 min] Channel: mycodeschool
  youtube.com/playlist?list=PL2_aWCzGMAwI3W_JlcBbtYTwiQSsOTa6P
- Playlist: Algorithms and Data structures [15 videos] [30 min] Channel: Gate Lectures
  youtube.com/playlist?list=PLEbnTDJUr_IeHYw_sfBOJ6gk5pie0yP-0
- Playlist: Design and Analysis of Algorithms [55 videos] [15 min] Channel: Computer Science and Engineering
  youtube.com/playlist?list=PLJ5C_6qdAvBE5VcLIv1xIFMRpGu3BQneh
- Playlist: Design and Analysis of Algorithms, Spring 2015 [35 videos] [80 min] Channel: MIT OpenCourseWare
  youtube.com/playlist?list=PLUl4u3cNGP6317WaSNfmCvGym2ucw3oGp
- Playlist: Introduction to Algorithms, Fall 2011 [45 videos] [50 min] Channel: MIT OpenCourseWare
  youtube.com/playlist?list=PLUl4u3cNGP61Oq3tWYp6V_F-5jb5L2iHb

Lecture Agenda

- ✓ Section 1: Data Structures and Algorithms Features
- ✓ Section 2: Data Structures and Algorithms Content
- ✓ Section 3: Practice on Online Judges
- ✓ Section 4: Programming Competitions
- ✓ Section 5: Tutorials and References
- ✓ Section 6: Online Courses


DO
MORE.