Goals

- To learn the main concepts and techniques of the algorithm design and analysis – the practical skills and basic theoretical basis
- To be able to choose, (design,) analyze and compare algorithms and data structures
- To learn to learn, use, solve, read, write, and present

Algorithms

- Al-Khwārizmī


Example

- Wind has blown away the +, *, (, ) signs
- What’s the maximal value?

\[ 2^671 \]

Q: How to maximize the value of any expression?

\[ 2 \times 45 \times 1981219872441123 = ? \]

13 x 11

\[
\begin{array}{c}
1 & 1 & 0 & 1 \\
\times & 1 & 0 & 1 \\
\hline
1 & 1 & 0 & 1 \\
1 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 \\
+ & 1 & 1 & 0 & 1 \\
\hline
1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
\end{array}
\]

(binary 143)
But Al-Khwārizmi knew another way to multiply, a method which is used today in some European countries. To multiply two decimal numbers a and b, write them next to each other, as in the example below. Then repeat the following: divide the first number by 2, rounding down the result (that is, dropping the .5 if the number was odd), and double the second number. Keep going till the first number gets down to 1. Then strike out all the rows in which the first number is even, and add up whatever remains in the second column.

\[
\begin{array}{c c c}
11 & 13 \\
5 & 26 \\
2 & 52 \text{ (strike out)} \\
\hline
1 & 104 \\
\end{array}
\]

\[
143 \text{ (answer)}
\]

---

**Analysis of algorithms**

- The theoretical study of computer-program performance and resource usage.
- What else is important than performance?
  - Modularity
  - Correctness
  - Maintainability
  - Functionality
  - Robustness
  - User-friendliness
  - Programmer time
  - Simplicity
  - Extensibility
  - Reliability

**Why study algorithms and performance?**

- Algorithms help us to understand scalability.
- Performance often draws the line between what is feasible and what is impossible.
- Algorithmic mathematics provides a language for talking about program behavior.
- Performance is the currency of computing.
- The lessons of program performance generalize to other computing resources.
- Speed is fun!

---

**Contents**

- Algorithms and Data structures – basics
- Abstract data types (+ STL, LEDA, ...)
- Lists, trees, heaps, ...
- Graphs (TSP; shortest paths, diameter, ...)

- Analysis of algorithm complexity
  - \( O(n \log n) \), \( o(n \log n) \), \( \Omega(n \log n) \), \( \Theta(n \log n) \)
  - recurrences \( T(n) = 2 T(n/2) + n \)
  - Turing machines, NP, NP-complete, PSPACE, ...
  - Very tough (NP) problems (“Monkey problem”; unsolvable problems?)
Contents...

- Basic designs: Dynamic programming, divide and conquer, randomised algorithms, ...
- Deterministic or non-deterministic
- Exact or approximate
- Heuristics (approximations):
  - Greedy Algorithms (packing, set cover, weighted set cover, ...) local search, tabu search
  - Simulated annealing (e.g. TSP),
  - Genetic algorithms (?), ...
  - Monte Carlo, Las Vegas

- Sorting, search, dictionary, ...
- Text algorithms
- Numerical algorithms
- Computational geometry
- Online algorithms vs offline
- Parallel algorithms

Skills/goals

- Bit-level understanding of low-level implementation
- Creative use of existing implementations
- Formal analysis, complexity theory
- Typical heuristics and algorithm design elements

- $67 + 56 = ?$
Programming languages:

- Pseudocode
  - primarily this is about pseudocode
- Pointers, arrays, memory handling
  - Low- and high level abstractions
- C/C++
- Java, Python, perl, ...
  - Explicit data structures and algorithms
- Whichever choice: you must be able to explain it
- Clarity and simplicity is a key

“The Textbook”

- Introduction to Algorithms, Second Edition
  Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein

- http://mitpress.mit.edu/algorithms/
- http://projects.csail.mit.edu/cla

  David Harel, Yishai Feldman
  Addison Wesley; 3 edition (June 11, 2004)

- Foundations of Computer Science: C Edition
  Alfred V. Aho, Jeffrey D. Ullman
  W. H. Freeman (October 15, 1994)
• The Art of Computer Programming (TAOCP)
  • Donald E. Knuth.

• The Algorithm Design Manual
  • *Steven S. Skiena*

• Algorithms [ILLUSTRATED]
  • Sanjoy Dasgupta, Christos Papadimitriou, Umesh Vazirani

---

**Mathematics**

• Concrete Mathematics: A Foundation for Computer Science (2nd Edition) (Hardcover)
  • Ronald L. Graham, Donald E. Knuth, Oren Patashnik
  • Hardcover: 672 pages
  • Addison-Wesley Professional;
  • 2 edition (March 10, 1994)

  • Robert Sedgewick

• Safari Bookshelf! (UT)

---

**More books**

• Parallel algorithms
  • An Introduction to Parallel Algorithms *Joseph Jaja*

• Heuristics
  • How to Solve It: Modern Heuristics. By Zbigniew Michalewicz, David B. Fogel
  • Edition 2. Illustrated Published by Springer, 2004

• Randomized algorithms

• Complexity theory
  • This book is a classic, developing the theory, then cataloguing many NP-Complete problems.

---

**Wikipedia:**


• ...