

- [Algorithm Design](#)
  - [1. Representative Problems](#)
    - [1.1 Stable Matching](#)
    - [1.2 Five Representative Problems](#)
  - [2. Algorithm Analysis](#)
    - [2.1 Computational Tractability](#)
    - [2.2 Asymptotic Order of Growth](#)
    - [2.3 Common Running Times](#)
  - [3. Graphs](#)
    - [3.1 Basic Definition](#)
    - [3.2 Connectivity and Traversal](#)
    - [3.3 Implementing Graph Search](#)
    - [3.4 Bipartiteness](#)
    - [3.5 Connectivity in Digraphs](#)
    - [3.6 Topological Order in DAGs](#)
  - [4. Greedy Algorithms](#)
    - [– Coin Changing](#)
    - [4.1 Interval Scheduling](#)
    - [4.2 Minimizing Lateness](#)
    - [4.3 Optimal Caching](#)
    - [4.4 Shortest Paths in a Graph](#)
    - [– Minimum Spanning Trees](#)
    - [– Prim, Kruskal, Borůvka](#)
    - [4.7 Single-Link Clustering](#)
    - [4.9 Min-Cost Arborescences](#)
  - [5. Divide and Conquer](#)
    - [5.1 Mergesort](#)
    - [5.3 Counting Inversions](#)
    - [5.4 Closest Pair of Points](#)
    - [– Randomized Quicksort](#)
    - [– Median and Selection](#)
    - [– Master Theorem](#)
    - [5.5 Karatsuba's Algorithm](#)
    - [– Strassen's Algorithm](#)
    - [5.6 Convolution and FFT](#)
  - [6. Dynamic Programming](#)
    - [6.1 Weighted Interval Scheduling](#)
    - [6.2 Segmented Least Squares](#)
    - [6.3 Knapsack Problem](#)
    - [6.4 RNA Secondary Structure](#)

- [6.5 Sequence Alignment](#)
- [6.6 Hirschberg's Algorithm](#)
- [6.7 Bellman–Ford Algorithm](#)
- [6.8 Distance Vector Protocol](#)
- [6.9 Negative Cycles](#)
- [7. Network Flow](#)
  - [7.1 Max-Flow and Min-Cut](#)
  - [7.2 Ford–Fulkerson Algorithm](#)
  - [7.3 Capacity-Scaling](#)
  - [– Shortest Augmenting Path](#)
  - [– Blocking Flow](#)
  - [– Unit Capacity Networks](#)
  - [7.4 Bipartite Matching](#)
  - [7.5 Disjoint Paths](#)
  - [7.6 Demands and Lower Bounds](#)
  - [7.7 Survey Design](#)
  - [7.8 Airline Scheduling](#)
  - [7.9 Image Segmentation](#)
  - [7.10 Project Selection](#)
  - [7.11 Baseball Elimination](#)
  - [7.12 Assignment Problem](#)
- [8. Intractability](#)
  - [8.1 Polynomial-Time Reductions](#)
  - [8.2 Vertex Cover](#)
  - [8.3 Independent Set](#)
  - [8.4 Set Cover](#)
  - [8.5 3-Satisfiability](#)
  - [8.6 Hamiltonian Cycle](#)
  - [8.7 3-Dimensional Matching](#)
  - [8.8 Graph 3-Colorability](#)
  - [8.9 Subset Sum](#)
  - [8.10 P vs. NP](#)
  - [8.11 NP-Completeness](#)
  - [8.12 co-NP](#)
- [9. PSPACE](#)
  - [9.1 PSPACE](#)
  - [9.2 Quantified SAT](#)
  - [9.3 Planning Problem](#)
  - [9.4 PSPACE-Complete](#)
- [10. Limits of Tractability](#)
  - [10.1 Small Vertex Covers](#)
  - [10.2 NP-Hard Problems on Trees](#)
  - [10.3 Circular Arc Coloring](#)
  - [– Bipartite Vertex Cover](#)
- [11. Approximation Algorithms](#)
  - [11.1 Load Balancing](#)
  - [11.2 Center Selection](#)
  - [11.3 Vertex Cover](#)
  - [11.4 Weighted Vertex Cover](#)
  - [11.5 Generalized Load Balancing](#)
  - [11.6 Knapsack Problem](#)
- [12. Local Search](#)
  - [12.1 Gradient Descent](#)
  - [12.2 Metropolis Algorithm](#)

- [12.3 Hopfield Neural Networks](#)
- [12.4 Maximum Cut](#)
- [11.5 Nash Equilibria](#)
- [13. Randomized Algorithms](#)
  - [13.1 Contention Resolution](#)
  - [13.2 Global Min Cut](#)
  - [13.3 Linearity of Expectation](#)
  - [13.4 Max 3-SAT](#)
  - [13.5 Universal Hashing](#)
  - [13.6 Chernoff Bounds](#)
  - [13.7 Load Balancing](#)
- Extra Topics
  - [Data Structures](#)
    - [Amortized Analysis](#)
    - [Binomial Heaps](#)
    - [Fibonacci Heaps](#)
    - [Disjoint Sets](#)
  - [Linear Programming](#)
    - [Simplex Algorithm](#)
    - [Linear Programming Duality](#)
    - [Ellipsoid Algorithm](#)

# Lecture Slides for Algorithm Design

These are a revised version of the lecture slides that accompany the textbook [Algorithm Design](#) by Jon Kleinberg and Éva Tardos. Here are the original and official version of the [slides](#), distributed by [Pearson](#).

TOPIC	SLIDES	READINGS	DEMOS
<b>Stable Matching</b> ( <i>Gale–Shapley</i> )	<a href="#">1up</a> · <a href="#">4up</a>	Chapter 1	<a href="#">Gale–Shapley</a>
<b>Algorithm Analysis</b> ( <i>big O notation</i> )	<a href="#">1up</a> · <a href="#">4up</a>	Chapter 2	<a href="#">binary search</a>
<b>Graphs</b> ( <i>graph search</i> )	<a href="#">1up</a> · <a href="#">4up</a>	Chapter 3	–
<b>Greedy Algorithms I</b> ( <i>basic techniques</i> )	<a href="#">1up</a> · <a href="#">4up</a>	Chapter 4	<a href="#">interval scheduling</a> <a href="#">interval partitioning</a>
<b>Greedy Algorithms II</b> ( <i>shortest paths and MSTs</i> )	<a href="#">1up</a> · <a href="#">4up</a>	Chapter 4	<a href="#">Dijkstra</a> <a href="#">Prim</a> , <a href="#">Kruskal</a> , <a href="#">Borůvka</a> <a href="#">Edmonds</a>
<b>Divide and Conquer I</b> ( <i>sorting and selection</i> )	<a href="#">1up</a> · <a href="#">4up</a>	Chapter 5	<a href="#">merging</a> <a href="#">quickselect</a>
<b>Divide and Conquer II</b> ( <i>integer and polynomial multiplication</i> )	<a href="#">1up</a> · <a href="#">4up</a>	Chapter 5	–
<b>Dynamic Programming I</b>	<a href="#">1up</a> · <a href="#">4up</a>	Chapter 6	–

(basic techniques)

<b>Dynamic Programming II</b> (sequence alignment, Bellman–Ford)	<a href="#">1up</a> · <a href="#">4up</a>	Chapter 6	–
<b>Network Flow I</b> (maximum flow theory)	<a href="#">1up</a> · <a href="#">4up</a>	Chapter 7	<a href="#">Ford–Fulkerson</a>
<b>Network Flow II</b> (maximum flow applications)	<a href="#">1up</a> · <a href="#">4up</a>	Chapter 7	–
<b>Network Flow III</b> (assignment problem)	<a href="#">1up</a> · <a href="#">4up</a>	Chapter 7	–
<b>Intractability I</b> (polynomial-time reductions)	<a href="#">1up</a> · <a href="#">4up</a>	Chapter 8	–
<b>Intractability II</b> (P, NP, and NP-complete)	<a href="#">1up</a> · <a href="#">4up</a>	Chapter 8	–
<b>Intractability III</b> (coping with intractability)	<a href="#">1up</a> · <a href="#">4up</a>	Section 10.2, 11.8	<a href="#">independent set</a> <a href="#">vertex cover</a>
<b>PSPACE</b> (PSPACE complexity class)	<a href="#">1up</a> · <a href="#">4up</a>	Chapter 9	–
<b>Limits of Tractability</b> (extending limits of tractability)	<a href="#">1up</a> · <a href="#">4up</a>	Chapter 10	–
<b>Approximation Algorithms</b> (approximation algorithms)	<a href="#">1up</a> · <a href="#">4up</a>	Chapter 11	<a href="#">list scheduling</a>
<b>Local Search</b> (Metropolis, Hopfield nets)	<a href="#">1up</a> · <a href="#">4up</a>	Chapter 12	–
<b>Randomized Algorithms</b> (randomized algorithms)	<a href="#">1up</a> · <a href="#">4up</a>	Chapter 13	–
<b>Data Structures I</b> (amortized analysis)	<a href="#">1up</a> · <a href="#">4up</a>	Chapter 17 (CLRS)	<a href="#">dynamic table</a>
<b>Data Structures II</b> (binary and binomial heaps)	<a href="#">1up</a> · <a href="#">4up</a>	Chapter 6 (CLRS, 2nd edition)	<a href="#">binary heap</a> <a href="#">heapify</a>
<b>Data Structures III</b> (Fibonacci heaps)	<a href="#">1up</a> · <a href="#">4up</a>	Chapter 19 (CLRS)	–
<b>Data Structures IV</b> (union–find)	<a href="#">1up</a> · <a href="#">4up</a>	<a href="#">Section 5.1.4</a> (Dasgupta et al.)	–
<b>Linear Programming I</b> (simplex algorithm)	<a href="#">1up</a> · <a href="#">4up</a>	(Chvátal)	–
<b>Linear Programming II</b> (linear programming duality)	<a href="#">1up</a> · <a href="#">4up</a>	(Chvátal)	–
<b>Linear Programming III</b> (ellipsoid algorithm)	<a href="#">1up</a> · <a href="#">4up</a>	Lecture notes ( <a href="#">Michel Goemans</a> )	–

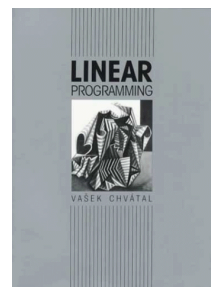
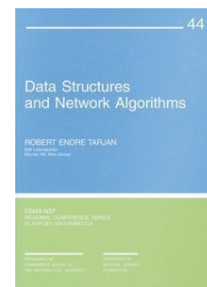
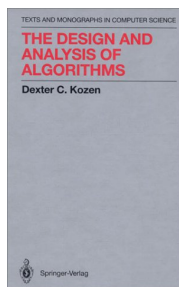
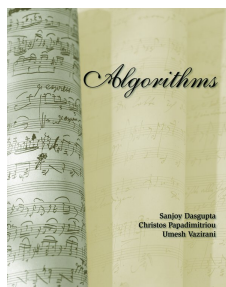
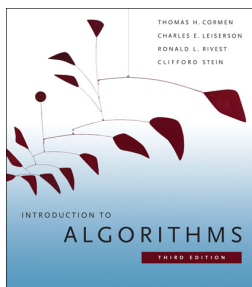
## References.

The lectures slides are based primarily on the textbook:

- [\*Algorithm Design\*](#) by Jon Kleinberg and Éva Tardos. Addison-Wesley, 2005.

Some of the lecture slides are based on material from the following books:

- [\*Introduction to Algorithms, Third Edition\*](#) by Thomas Cormen, Charles Leiserson, Ronald Rivest, and Clifford Stein. MIT Press, 2009.
- [\*Algorithms\*](#) by Sanjoy Dasgupta, Christos Papadimitriou, and Umesh Vazirani. McGraw Hill, 2006.
- [\*The Design and Analysis of Algorithms\*](#) by Dexter Kozen. Springer, 1992.
- [\*Algorithms 4/e\*](#) by Robert Sedgewick and Kevin Wayne. Addison-Wesley Professional, 2011.
- [\*Data Structures and Network Algorithms\*](#) by Robert Tarjan. Society for Industrial and Applied Mathematics, 1987.
- [\*Linear Programming\*](#) by Vášek Chvátal. W. H. Freeman, 1983.



## Instructors.

If you are an instructor using the textbook and would like the latest version of the keynote source files, please [email Kevin Wayne](#).

## Errata.

Here are the known [errata](#) in these lecture slides.

## Credits.

Special thanks to Pierre Flener, for finding and reporting dozens of errors and suggesting numerous improvements in the presentation.