

Adventures in Advanced Algorithms:

A Guide for Curious Students

Sepehr Assadi

University of Waterloo

© 2026 Sepehr Assadi • CC BY-NC-SA 4.0

September 17, 2026



© 2026 Sepehr Assadi. Some rights reserved. Released as an open educational resource under a [Creative Commons Attribution–NonCommercial–ShareAlike 4.0 International](#) license (CC BY-NC-SA 4.0): you may share and adapt this material for **non-commercial** purposes, with attribution, provided that any adaptations are distributed under these same terms.

Created and maintained by [Sepehr Assadi](#), School of Computer Science, University of Waterloo.

How to cite this book.

Sepehr Assadi. *Adventures in Advanced Algorithms: A Guide for Curious Students*. 2026. Open educational resource, sepehr.assadi.info/advanced-algorithms.

Contents

Preface	1
I Background and Motivating Examples	3
1 Motivating Example	4
1.1 Problem: $(\Delta + 1)$ Vertex Coloring	4
1.2 A Faster $(\Delta + 1)$ Coloring Algorithm	5
1.2.1 The Algorithm	6
1.2.2 Runtime Analysis	7
Exercises	10
2 Probabilistic Analysis I: Concentration Inequalities	12
2.1 Balls and Bins Experiments	12
2.1.1 Concentration Inequalities	13
2.1.2 Concentration Inequalities for Balls and Bins	14
2.1.3 Optimal Bounds for Balls and Bins?	17
2.2 Success Amplification of Randomized Algorithms	20
2.3 Proof of (a Special Case of) Chernoff Bound	21
3 Probabilistic Analysis II: Random Graph Theory	23
3.1 Power of Two Choices in Balls and Bins	23
3.1.1 Proof of Theorem 3.2 (assuming Lemmas 3.4 and 3.5)	25
3.2 Random Graph Theory: Proofs of Lemmas 3.4 and 3.5	26
3.2.1 Lemma 3.4 : Size of Connected Components in Random Graphs	27
3.2.2 Lemma 3.5 : Induced Edge Counts in Random Graphs	28
4 Streaming Algorithms I: Distinct Elements	30
4.1 Streaming Distinct Elements Problem	30

4.1.1	The Algorithm	31
4.1.2	Formal Analysis	33
4.2	Independence for Space: Hash Functions	35
4.2.1	Improving Space Complexity of Algorithm 4.4	36
4.3	The Original Distinct Elements Problem?	37
	Exercises	38
5	Streaming Algorithms II: Frequency Moments	40
5.1	Frequency Moments Estimation	40
5.2	First Frequency Moment: Morris Counter	41
5.3	Second Frequency Moment: AMS Sketch	44
II	Linear and Semidefinite Programming	48
6	Linear Programming I: Basics and Applications	49
6.1	Introduction to Linear Programming	49
6.1.1	Optimization Problems	49
6.1.2	Linear Optimization (a.k.a. Linear Programming)	50
6.2	Applications of Linear Programming	52
6.2.1	An Application of Type (a): Creating a “Balanced” Diet	52
6.2.2	Another Application of Type (a): Linear Regression	53
6.2.3	An Application of Type (b): The Bipartite Matching Problem	55
6.2.4	Another Application of Type (b): The Bipartite Vertex Cover Problem	57
	Exercises	59
7	Linear Programming II: LP Duality	60
7.1	A Motivating Example for Duality	60
7.2	Duality in Linear Programming	62
7.2.1	Primal and Dual LPs	62
7.2.2	Weak and Strong Duality Theorems	63
7.3	LP Duality Applications	65
7.3.1	Optimizing vs Feasibility Checking LPs	65
7.3.2	$\text{LP} \in \text{NP} \cap \text{coNP}$	65
7.3.3	Bipartite Matching and Bipartite Vertex Covers	66

8	Linear Programming III: Approximation Algorithms and Integrality Gaps	68
8.1	Integrality Gaps, Rounding, and Approximation Algorithms	68
8.1.1	Minimum Vertex Cover and Its LP	68
8.2	The Set Cover Problem	70
8.3	Randomized Rounding for Set Cover	71
8.4	Dual Fitting for Set Cover	73
	Exercises	76
9	Linear Programming IV: A Primal-Dual Algorithm for Bipartite Matching	77
9.1	Bipartite Matching and Vertex Cover Problems	77
9.1.1	Warm-Up: A Half-Approximation Algorithm	78
9.2	A Primal-Dual Algorithm for $(1 - \varepsilon)$ -Approximation	79
9.2.1	The Analysis	80
9.2.2	Runtime	82
9.3	An Exact Algorithm for Bipartite Matching	82
	Exercises	84
10	Linear Programming V: Center of Gravity Algorithm	85
10.1	Basics of Convexity in Linear Programming	85
10.1.1	Volume of Convex Sets and Center of Gravity	87
10.2	Center of Gravity Algorithm	88
10.3	Communication Complexity of Bipartite Matching	90
10.4	Another Application: Submodular Minimization	93
	Exercises	93
11	Semidefinite Programming I: Maximum Cut	94
11.1	Recap on Linear Algebra	94
11.2	Semidefinite Programming	95
11.3	The Maximum Cut (Max-Cut) Problem	96
11.3.1	An Easy Algorithm and a Linear Program	96
11.3.2	Quadratic Programming	98
11.4	A Semidefinite Programming Approach for Max-Cut	98
11.4.1	The Algorithm for Max-Cut	100
12	Semidefinite Programming II: Vertex Coloring	103
12.1	The 3-Coloring Problem	103

12.1.1	Independent Sets and Coloring	103
12.2	Finding Large Independent Sets in 3-Colorable Graphs	104
12.2.1	A Simple Randomized Algorithm	104
12.2.2	A Better Algorithm Tailored to 3-Colorable Graphs	105
12.3	Finding Independent Sets via SDPs	106
12.3.1	An SDP for 3-Colorable Graphs	106
12.3.2	Rounding the SDP via Random Separators	107
III	Multiplicative Weight Update Technique	112
13	Multiplicative Weight Update I: Learning from Experts	113
13.1	The Experts Problem	113
13.1.1	Warm-Up 1: An <i>Ideal</i> Expert	114
13.1.2	Warm-Up 2: A Crude Approximation	114
13.2	The Weighted Majority Algorithm	115
13.3	The Multiplicative Weight Update Algorithm	117
14	Multiplicative Weight Update II: Approximating Linear Programs	119
14.1	Multiplicative Weight Update for the Matching LP	119
14.1.1	The Setup and the Oracle LP	119
14.1.2	The Algorithm	121
14.1.3	The Analysis	122
14.1.4	Runtime Analysis	124
15	Multiplicative Weight Update III: Width Reduction Techniques	126
15.1	A Faster MWU Algorithm for Bipartite Matching	126
15.1.1	A Strategy for Improved Algorithms	127
15.1.2	Reducing the Width: Approach One	129
15.1.3	Reducing the Width: Approach Two	130
15.1.4	A (Slightly) Better Algorithm for Additive Approximation	131
	Exercises	133
16	Multiplicative Weight Update IV: Undirected Maximum Flow	134
16.1	The Maximum Flow Problem	134
16.1.1	Maximum Flow via MWU	135

16.1.2	The Plan for the Maximum Flow Algorithm	136
16.2	Next Attempt: A Stronger Oracle for Maximum Flow	137
16.2.1	Detour: Electrical Flows	138
16.2.2	Back to the Oracle	139
16.3	An Even Stronger Oracle for Maximum Flow	141
16.3.1	Detour: More on Electrical Flows	141
16.3.2	Back to the Oracle	142
17	Multiplicative Weight Update V: Adaptive Sparsification	147
17.1	This chapter is under construction	147
IV	Graph Algorithms	148
18	Minimum Spanning Trees I: A Fast Deterministic Algorithm	149
18.1	Minimum Spanning Trees	149
18.1.1	The Basic Rules of MSTs	150
18.2	Refresher on the Classical Algorithms for MSTs	150
18.2.1	Kruskal's Algorithm	151
18.2.2	Prim's Algorithm	152
18.2.3	Boruvka's Algorithm	153
18.3	Fredman-Tarjan Algorithm in $O(m \log^*(n))$ Time	154
18.3.1	Warm-Up: Improving Prim's Runtime via Fibonacci Heaps	154
18.3.2	The Fredman-Tarjan Algorithm	154
	Exercises	157
19	Minimum Spanning Trees II: A Linear-Time Randomized Algorithm	158
19.1	A Linear Time Randomized Algorithm for MST	158
19.1.1	Preliminaries	158
19.1.2	The KKT Algorithm	159
19.2	Detour: Optimal Algorithms with Unknown Runtime?	162
	Exercises	163
20	Shortest Paths I: Spanners and Probabilistic Tree Embeddings	164
20.1	Graph Spanners	164
20.1.1	Density of Graphs without Short Cycles	166

20.2 Asymptotic Optimality of Proposition 20.5 for $k = 2$	167
20.2.1 A “Weak” Construction Using Random Graphs	167
20.2.2 An Asymptotically Optimal Construction	168
20.3 Probabilistic Tree Embeddings	170
20.4 Low Diameter Decomposition (LDD)	170
20.5 Back to Tree Embeddings: Proof of Theorem 20.8	173
21 Shortest Paths II: Negative-Weights in Nearly-Linear Time	175
21.1 Negative Weight (Single-Source) Shortest Paths	175
21.2 Background Tools	176
21.2.1 Price Function	176
21.2.2 Reducing Out Degrees	177
21.2.3 Further Properties of Bellman-Ford and Dijkstra’s Algorithms	178
21.2.4 Low Diameter Decomposition for Directed Graphs	180
21.3 An $\tilde{O}(m\sqrt{n})$ Time Algorithm for SSSP	180
21.3.1 Proof of Lemma 21.17 : The Scaling Lemma	181
21.4 The Final $\tilde{O}(m \log W)$ Time Algorithm	185
Exercises	188
22 Edge Coloring: A Near-Linear Time Algorithm	189
22.1 This chapter is under construction	189
V More Advanced Probabilistic Techniques	190
23 Probabilistic Method and Lovász Local Lemma	191
23.1 Motivation	191
23.1.1 Ramsey numbers	191
23.1.2 Large Independent Sets in Bounded-Degree Graphs	193
23.1.3 Coloring 3-Uniform Hypergraphs	194
23.2 Lovász Local Lemma	195
23.2.1 Application 1: Independent Sets in Bounded-Degree Graphs	196
23.2.2 Application 2: Coloring 3-Uniform Hypergraphs	196
Exercises	197
24 Algorithmic Lovász Local Lemma and Entropy Compression	198

24.1	Proof of Lovász Local Lemma	198
24.2	An <i>Algorithmic</i> LLL?	200
24.2.1	An Algorithmic Version of Proposition 24.2	201
24.2.2	Entropy Compression Method and Runtime Analysis of Algorithm 24.3	201
	Exercises	204
25	Strong(er) Concentration Bounds I: Martingales and Azuma's Inequality	206
25.1	Balls and Bins Revisited	206
25.2	Strengthening the Chernoff Bound	207
25.2.1	Conditional Expectations	207
25.2.2	Martingales	208
25.2.3	Azuma's Inequality	210
25.3	Applications	210
25.3.1	Application 1: Balls and Bins	210
25.3.2	Application 2: Chromatic Number	212
	Exercises	216
26	Strong(er) Concentration Bounds II: Talagrand's Inequality	218
26.1	Coloring Locally Sparse Graphs	218
26.1.1	Proof of Lemma 26.4	219
26.1.2	Concluding the Proof of Theorem 26.2	222
	Exercises	223
27	Strong(er) Concentration Bounds III: Gaussian Variables	225
27.1	String Similarity	225
27.1.1	Attempt 1: Random indices	225
27.2	Johnson-Lindenstrauss Lemma (JLL)	226
27.2.1	Attempt 2: Gaussians	226
27.3	Subspace embedding	229
27.3.1	γ -Nets	230
27.3.2	A (Weaker) One Shot Argument via ε -Nets	231
27.3.3	The Optimal Bound: Chaining via $(1/2)$ -Nets	233
27.4	A Simple Application: ℓ_2 -Regression	235
	Exercises	236
28	Polynomial Methods: Schwartz-Zippel Lemma	238

28.1 Fingerprinting	238
28.1.1 Public Randomness	238
28.1.2 Error-Correcting Codes	238
28.1.3 Hash Functions and Prime Argument	238
28.1.4 Polynomial Argument	239
28.2 Matrix Multiplication	239
28.3 Polynomial Identity Testing	240
28.4 Matchings	242
28.5 Proof of Schwartz-Zippel Lemma	244
28.6 A Parallel Algorithm for Bipartite Matching	245
28.6.1 Isolation Lemma	246
28.6.2 Isolation Lemma for Parallel Matching	246
28.7 Arthur-Merlin Communication	247
Exercises	249
29 Random Walks I: Markov Chains	251
29.1 Motivating Examples	251
29.1.1 A Randomized Algorithm for 2-SAT	251
29.1.2 Undirected s - t Connectivity (USTCONN) with Small Space	253
29.2 Markov Chains	254
29.2.1 Hitting Times and Stationary Distributions	255
29.3 Random Walk on Undirected Graphs	257
Exercises	261
30 Random Walks II: Graphs and Hitting Times	263
30.1 A Quick Recap of Random Walks	263
30.2 Perfect Matching in Regular Bipartite Graphs	263
30.2.1 Role of Augmenting Paths	265
30.2.2 The Algorithm of [GKK13] for Regular Bipartite Graphs	266
30.2.3 Finding Augmenting Paths	267
Exercises	270
31 Random Walks III: Mixing Time and Couplings	272
31.1 Convergence to Stationary Distribution: Mixing Time	272
31.1.1 Detour: Total Variation Distance	272

31.1.2	Mixing Time in Random Walks	275
31.1.3	A Motivating Application: Sampling Colorings	275
31.2	Coupling and Bounding Mixing Times	277
31.2.1	Detour: Coupling	277
31.2.2	Using Coupling for Bounding Mixing Times	278
31.2.3	Conclusion: Mixing Time for Sampling Colorings	280
	Exercises	281
VI	Modern Models of Computation	284
32	Graph Streaming I: Maximum Matching in a Single Pass	285
33	Graph Streaming II: Sketching Connectivity and Minimum Cut	286
33.1	Graph Sketching: Spanning Forests	286
33.1.1	Attempt 1: A “too-good-to-be-true” approach	287
33.1.2	Attempt 2: The Approach	291
34	Graph Streaming III: Shortest Path in Multiple Passes	292
35	Online Algorithms I: Online Bipartite Matching	293
35.1	The Online Matching Problem	293
35.1.1	Deterministic Algorithms?	293
35.1.2	Randomized Algorithms?	294
35.1.3	The RANKING Algorithm	295
35.2	A Primal-Dual Analysis of the RANKING Algorithm	296
36	Online Algorithms II: Secretary Problem and Prophet Inequalities	301
37	Online Algorithms II: Secretary Problem and Prophet Inequalities	302
37.1	Online Decision Making	302
37.2	The Secretary Problem	303
37.3	The Prophet Inequality	305
38	Distributed Algorithms I: Vertex Coloring and Shattering	308
38.1	Local Computation Algorithms	308
38.1.1	Generic Randomized $\Delta + 1$ Coloring	309
38.1.2	Shattering Arguments	311

38.2 Maximal Independent Sets (MIS)	313
38.2.1 Random-order MIS	313
38.2.2 Runtime Analysis	313
39 Distributed Algorithms II: Distributed Bellman-Ford Algorithm	317
A Probabilistic Background	318
B Cheat Sheet	322
Bibliography	323