

Algorithms to Live by - The Computer Science of Human Decisions

My Thoughts

Algorithms to Live By is a surprisingly fun book considering the subject. Contains **mathematical philosophy** on decision making on a wide range of topics. Including hiring, dating, real estate, sorting, and even doing laundry.

My Favorite Quotes

- *Inaction is just as irrevocable as action.*
- *Instead of thinking about only the next decision you will make, **think about all the decisions** you are going to make about the same options in the future.*
- *Seizing the **day** and seizing a **lifetime** are two entirely different endeavors.*
- *When balancing favorite experiences and new ones, nothing matters more than the **interval** over which we **plan to enjoy** them.*
- ***Explore** when you will **have time** to use the resulting knowledge; **exploit** when you are **ready to cash-in**.*
- ***Exploration** in itself has value.*
- *Sorting something you will never search for is a complete waste. Searching for something you never sorted is merely inefficient.*
- ***Forgetting** can be as important as remembering.*
- *How we **spend our days** is how we **spend our lives**.*
- *The best strategy for getting things done **might be to slow down**.*
- *If you can't **explain it simply**, you don't **understand it well enough**.*
- *The right action can produce a bad outcome. **Process is all we have control over**, not results.*
- *We can hope to be fortunate, but we should **strive to be wise**.*

Introduction - Algorithms and topics covered

1. Optimal stopping
2. Explore/exploit
3. Sorting theory
4. Caching theory
5. Scheduling theory
6. Game theory

Chapter 1: Optimal Stopping - *When to stop looking.*

*“37% rule. Gather data for **the first 37%**, then **make a decision** (leap) as soon as you find an **option better** than the first 37%.”*

*“Apartment hunting, **spend 37% of total search time** (11 of 30 days for example) looking at apartments to establish a baseline **before making a decision.**”*

Secretary problem. *The Merrill Flood was discovered in 1958.*

*“Two ways to fail, **stopping early and stopping late.**”*

37% can be applied to the number of applicants or to **the amount of time.**

*Selecting a mate, using age 18-40 as the time frame, **26.1 years old** is the optimal point to switch from looking to leaping.*

If second chances are allowed, with 50% chance of yes on second ask, then optimal looking point is 61% with 61% chance of success.

Variables: rejection and recall.

Knowing a good thing when you see it: **full information.**

One setup: “no information” games. Starting with no info and no baseline.

Full information. *Knowing what the perfect applicant is.* No need to look before leaping. **Setup a threshold rule.** Immediately **decide/leap** (hire) if **above a certain percentile.**

Decision based on how many applicants remain. Raise and lower the threshold based on how many applicants remain.

58% chance of hiring the best applicant under full information setup.

When to Sell (Real Estate)

Similar to full information game. We objectively know the value of each offer and the market value. Waiting has a cost measured in dollars. A good offer today beats a slightly better offer several weeks from now. Set a threshold going in, ignore every offer below, and immediately accept any offer above.

Cost benefit analysis of the waiting game. If there is a risk of offers or savings running out.

When to Quit - Inaction is just as irrevocable as action.

Chapter 2: Explore/Exploit

The **latest** vs the **greatest**.

- Explore = gathering information
- Exploit = using information

Multi-armed bandit problem: (scenario) derived from name of a casino slot machine “one armed bandit.”

Example: comparing one machine you won **9 of 15 pulls** vs another that you won **1 of 2 pulls**.

Our goals should change as we age.

“Instead of thinking about only the next decision you will make, think about all of the decisions you are going to make about the same options in the future.”

How long do you plan to “be in the casino” impacts the answer.

Seize the Interval

Seizing the day and seizing a lifetime are two entirely different endeavors.

When balancing favorite experiences and new ones, nothing matters more than the interval over which we plan to enjoy them.

More likely to try a new restaurant when entering a city than when leaving.

The **value of exploration** (finding a new favorite) **can only go down** over time as the remaining opportunities to savor it dwindle.

The flip side, **the value of exploitation** can only **go up over time**.

Explore when you will have time to use the resulting knowledge; exploit when you are ready to cash-in.

The interval makes the strategy. - By observing the strategy, we can also infer the interval.

Exploration in itself has value.

Regret and Optimism

Upper confidence bound algorithms.

Optimism is the best prevention for regret.

Childhood is the optimum time to explore without worry of payout.

Chapter 3: Sorting

The Ecstasy of Sorting

Herman Hollerith invented the first Sorting machine in 1880s and eventually became IBM in 1911. The machine was used to sort census cards in the 1890 census.

Sorting spurred the development of computers.

Search engines are more like sort engines.

Sorting is pleasing to the eye.

The Agony of Sorting

With sorting, **size is a recipe for disaster**.

The first and most fundamental rule of sorting: scale hurts.

Record for sorting a deck of 52 cards is 36 seconds.

Determine how you are going to measure, best case scenario time or average sort time.

Also need to know worst time or worst case scenario.

This chapter and book is discussing worst case scenario unless noted otherwise.

Computer science short hand term is “Big O” notation for algorithmic worst case scenarios.

Sheds fine details, schema for dividing problems into different broad classes.

- **Big O of “1” (Constant Time)**

Example, the time it takes to clean your house before a party. The time is always the same, totally invariant of the guest list.

Same amount of work regardless of the number of people who attend.

- **Big O of “N” (Linear Time)**

Time required to pass the roast around the table.

Twice the guests requires twice the time.

- **Big O of “N Squared” (Quadratic Time)**

Each guest arriving at the party hugs each person.

First person hugs you, second person hugs you and the first guest (2 hugs), third person hugs you and both guests (3 hugs), etc.

- **Big O of “2 to the N” (Exponential Time)**

Where each additional guest doubles your work.

- **Big O of “N Factorial” (Factorial Time)**

A class of problems only joked about by computer scientists.

The Squares: Bubble Sort and Insertion Sort

Bubble sort is simple but extremely inefficient. Quadratic time.

Bubble sort is scanning over the line and comparing two side by side and moving one to the left or right, going back over the shelf over and over until it is sorted. Example is sorting a bookshelf of books in alphabetical order. The problem is each scan through the shelf only allows you to move each book one position at most.

Insertion Sort: take every book off the shelf and put them back on one at a time. Runs a bit faster than bubble sort. Quadratic time.

What is the minimum effort of time required to create order?

Merge sort is between Linear Time and Quadratic Time, one of the legendary algorithms in computer science.

Merge sort is the divide and conquer approach. You can collate two sorted stacks almost instantly.

In sorting a census level number of items, this is a difference between making 29 passes through the data set and 300 million.

Method of choice for large scale industrial sorting problems.

Can easily be paralleled.

Beyond comparison, outsmarting the logarithm.

Preston sort center, one of the biggest and most efficient book sorting facilities in the world.

Run by King County Library.

Bucket Sort

Items grouped into a number of general categories.

Sorting is prophylaxis for searching.

Central trade-off between sorting and searching.

The effort expended on sorting materials is a preemptive strike against the effort to search them later.

Sorting something you will never search for is a complete waste. Searching for something you never sorted is merely inefficient.

Google for example, presorts search results by machine so that searching is done in seconds. Most domestic bookshelves do not need to be sorted.

Comparison: Counting Rank vs Sort

Blood Sort

Hierarchy

Online poker

Animals

A race is fundamentally different than a fight.

Constant Time Algorithm

Assign cardinal numbers instead of ordinal

Chapter 4: Caching: Forget About It

"Forgetting can be as important as remembering."

The Memory Hierarchy

Computer hard drive vs solid state drive.

A small fast memory and a large slow one.

Computer memory access has not increased as fast as processing speed.

Most computers, phones and tablets have a six layer memory hierarchy.

What do we do when memory gets full?

Eviction and clairvoyance.

There comes a time when for every addition of knowledge you forget something that you knew before.

Eviction policy. Caching algorithm.

Known today as Bellamy's Algorithm.

Approach options to managing the cache:

Random.

FIFO

LRU = Least Recently Used. Evict the item that has gone the longest untouched.

LRU method consistently performed the best.

Temporal locality.

The last thing we will likely need is the thing we have gone longest without.

History repeats itself backwards.

Our best guide to the future is a mirror image of the past.

Caching physical items like library books, internet servers and files, Amazon warehouse items, etc.

Multi-level memory hierarchy.

Self-organizing lists. Research paper.

Always put an item back at the front of the list, this utilizes the LRU principle.

The Forgetting Curve

Ebbinghaus study.

A big book is a big nuisance.

Forgetting things and taking longer to process is largely a result of knowing more and having more memories to process as we age and get older.

Chapter 5: Scheduling

*“First Things First - How we spend our days is how we spend our lives.
We are what we repeatedly do. Aristotle”*

Laundry: start with the fastest wash and end with the fastest dry. This gives maximum overlap.
Two machine scheduling (washer and dryer)

Single machine scheduling (yourself)

Handling Deadlines - Make your goals **explicit**.

Strategies:

Sort by earlier due date. Optimal strategies for reducing maximum lateness.

Minimizing the number of items late: optimal is using Moore’s Algorithm.

Getting things done; **get as many things done as quickly as possible**. Calculate using the *sum of completion times* method.

Weighted completion times. Divide weight of each task by the time to complete. Work by highest result per unit of time. This reduces the total weight.

Priority Inversion and Precedence Constraints

Priority inheritance.

Most scheduling problems are intractable.

Preemption and Uncertainty

Thrashing

Interrupt Coalescing

Tension between responsiveness and throughput.

The best strategy for getting things done might be to slow down.

Chapter 6: Bayes Rule

Predicting the future.

Reverend Thomas Bayes

Hypothetical reasoning forward allows us to reason backwards to solve problems.

The Copernican Principle

Predicting you have arrived at any point in time at the mid-point. Longevity of Berlin Wall example.

Good principle when we have nothing to go on for estimating.

Bayes meets Copernicus

Real World Priors

Two types of things:

Things that tend towards and cluster around a natural value (human lifespan).

Things that don't.

Bell curve distribution.

Power law distribution (town population average). Most things below the mean and a few enormous ones above it.

Erlang Distribution

Totally variant results.

Three types of rules for predicting;

Multiplicative

Average

Additive

Know what type of distribution you are up against.

Chapter 7: Over-fitting

When to think less.

Pro and con lists. Recommended by Benjamin Franklin.

There can be wisdom to deliberately thinking less in specific circumstances.

Cross validate to prevent over fitting.

Use secondary data points to check the first data point.

How to combat over fitting.

Penalizing complexity.

If you can't explain it simply, you don't understand it well enough.

Occam's Razor.

9 factor model vs 3 factor model.

Allowing more time can create more complexity and be counterproductive.

Early stopping.

When to think less.

When you have high uncertainty and limited data.

Chapter 8: Relaxation

Let it slide.

Constrained optimization problems.

Known as the traveling salesman problem.

Circuit lawyer traveling to different cities trying to determine optimum route.

Traveling salesman problem is currently intractable (unsolvable). The next closest answer that was easily solvable is the minimum spanning tree which is the minimum distance connecting all points (cities).

Constraint relaxation.

Try solving an easier version of the problem first, by relaxing the constraints.

Discrete optimization problems. You are either seated at table A or B, no in between.

Placing fire stations optimally in a city.

Lagrangian relaxation.

Two parts of an optimization problem; the rules and the score keeping.

This is how sports schedules are put together.

Napsack Problem.

Chapter 9: Randomness

When to leave it to chance.

Randomized algorithms.

The Monte Carlo method.

Probability of winning solitaire.

Sometimes sampling by playing/trying is better than a mathematical solution.

Algorithms to determine prime numbers. Prime numbers are used for cryptography.

Rabin's Algorithm.

Randomness is the best way of testing certain problems. Like Polynomial Identity test.

Use sampling of random numbers for X to test results.

The Three Part Trade-off

Time

Space

Error probability

Bloom filter.

Metropolis Algorithm.

Simulated Annealing.

Chapter 10: Networking

How we connect.

Communication is by protocol.

TCP protocol.

Packet switching vs old phone style circuit switching.

Exponential-back off: the algorithm of forgiveness.

Flow control and congestion avoidance.

AIMD = additive increase, multiplicative decrease.

Buffer bloat.

Dropped packets or dropping the ball.

Chapter 11: Game Theory

The minds of others.

Literary plots usually belong to one of these categories:

- Man vs Nature
- Man vs Self
- Man vs Man
- Man vs Society

This is man vs man and man vs society.

Algorithmic game theory.

Recursion

The halting problem.

The goal is to play only one level above your opponent.

Poker players.

Leveling war.

Nash equilibrium.

Rock Paper Scissors.

Dominant Strategies

The prisoners dilemma.

The Tragedy of Commons

Mechanism Design: change the game.

Change the game instead of the strategy.

Sometimes called reverse game theory.

Ask what rules will create the behavior we want.

Example: prisoners dilemma with the Godfather forcing them to be loyal and not inform on each other.

Information cascade.

Vickrey sealed-bid auction.

Conclusion: computational kindness

The right action can produce a bad outcome. Process is all we have control over, not results.

We can hope to be fortunate but we should strive to be wise.

Stating your preferences helps reduce the computational social problem. For example your preference for where to eat dinner.

Making people infer your preferences puts more computational pressure on the group.