jeremy kubica

THE CS DETECTIVE

An Algorithmic Tale of Crime, Conspiracy, and Computation

THE CS DETECTIVE

An Algorithmic Tale of Crime, Conspiracy, and Computation

jeremy kubica

no starch press

SAN FRANCISCO

Printed in USA

First printing

20 19 18 17 16   1 2 3 4 5 6 7 8 9

ISBN-10: 1-59327-749-0
ISBN-13: 978-1-59327-749-9

Publisher: William Pollock
Production Editor: Riley Hoffman
Cover and Interior Design: Beth Middleworth
Illustrator: Miran Lipovača
Developmental Editor: Liz Chadwick
Technical Reviewer: Heidi Newton
Copyeditor: Rachel Monaghan
Compositor: Riley Hoffman
Proofreader: Paula L. Fleming

For information on distribution, translations, or bulk sales, please contact No Starch Press, Inc. directly:

No Starch Press, Inc.
245 8th Street, San Francisco, CA 94103
phone: 415.863.9900; info@nostarch.com
www.nostarch.com

Library of Congress Cataloging-in-Publication Data
Names: Kubica, Jeremy.
Title: The CS detective : an algorithmic tale of crime, conspiracy, and computation / by Jeremy Kubica.
Description: San Francisco : No Starch Press, [2016] | Summary: "A mystery novel for computer science students and enthusiasts that introduces the concepts behind search algorithms and data structures. Each chapter teaches a new concept, ending with a technical explanation."-- Provided by publisher.
Identifiers: LCCN 2016013632 (print) | LCCN 2016026435 (ebook) | ISBN 9781593277499 (pbk.) | ISBN 1593277490 (pbk.) | ISBN 9781593277871 (epub) | ISBN 1593277873 (epub) | ISBN 9781593277888 (mobi) | ISBN 1593277881 (mobi)
Subjects: | CYAC: Algorithms--Fiction. | Computer science--Fiction. | Mystery and detective stories.
Classification: LCC PZ7.1.K8 Cs 2016 (print) | LCC PZ7.1.K8 (ebook) | DDC [Fic]--dc23
LC record available at https://lccn.loc.gov/2016013632

CONTENTS

Acknowledgments

I am tremendously grateful to all the people who contributed to, worked on, and supported this book.

I would like to start by thanking the whole team at No Starch Press. In particular, I would like to thank Liz Chadwick and Riley Hoffman for their excellent help, guidance, and suggestions during the editing process. Liz's brilliant suggestions were invaluable for keeping the story focused and moving. I am also grateful for her ideas in shaping the lecture notes format of the technical sections. Thank you to Bill Pollock and Tyler Ortman for their support, and a special thanks to Bill for suggesting the title. I would also like to thank Carlos Bueno for pointing me to No Starch Press.

Thank you to Miran Lipovača for his amazing illustrations. They truly capture the characters and the story.

Thank you to Heidi Newton for her thorough and insightful technical review. Her work was vital in ensuring the concepts came across both accurately and understandably. I appreciate the number of times she warned of explanations being too technical and thus inaccessible to students.

A tremendous thanks goes out to all the people who slogged through earlier versions of this book and provided valuable feedback: John Bull, Mike Hochberg, Edith Kubica, Regan Lee, and Kristen “Kit” Stubbs, PhD. Thank you to Ilana Schwarcz, who edited the earlier version of the manuscript and helped smooth out many of the rough edges.

A deep thank you to my family for their support. In particular, thank you to my parents for supporting my interest in computer science as a kid and for providing significant encouragement for this book.

A Note to Readers

This book focuses on computational thinking and search algorithms. The stories introduce and illustrate computational concepts at a high level, exploring the motivation behind them and their application in a noncomputer domain. This book is not a comprehensive text, and the stories are not intended as a substitute for a solid technical description of computer science. Instead, they are meant to be used like illustrations: they supplement the full concept and aid understanding.

The book covers a variety of computational approaches that all share the broad categorization of *search algorithms.* Concepts are presented first within the context of the story, and then explained more technically in a section laid out as lecture notes at the end of each chapter. Readers can safely skip these technical sections without missing any of the story.

This book assumes some experience with basic computer science concepts but does not require knowledge of any specific programming language. The algorithms in this book are meant to apply to a range of programming languages and problem domains.

— 1 —

Search Problems

The door opened without a knock—only the hinge's creak announced the visitor. Frank started for his crossbow, but pulled up short. If the Vinettees were coming for him, they would have knocked—with an axe. Whoever was coming through the door must want to talk. Frank reached for his mug instead and downed the remainder of his now-cold coffee.

"Captain Donovan," he said as the man entered. "What brings you to this fine neighborhood? I thought you didn't venture below Fifteenth Street anymore."

"It's been a while," the captain said simply. "How've you been, Frank?"

"Spectacular," Frank answered dryly, eyeing the captain as he walked a slow circuit around the room.

Donovan scanned Frank's shabby office. His red officer's cloak swished gently behind him. "How's the private eye game?"

"It pays the bills," Frank lied.

The captain nodded. He paused for a moment, then moved to the bookshelf and browsed the contents.

"So is this a social visit then?" Frank said. "Should I be asking after Marlene and the kids?"

“They’re quite well,” replied Donovan without turning around. “Marlene’s turtle-grooming business is doing well these days. Bill joined the force last year. And Veronica is an accountant, just about the last thing we would have—”

“I wasn’t actually asking,” Frank interrupted.

The captain shrugged. He pulled a book from the shelf and leafed through the pages. Frank craned his neck to see the cover—*Police Academy Yearbook: Class XXI.*

“What do you want, Captain?” Frank demanded.

The captain met Frank’s stare at last. “I need your help, Frank,” he said.

Frank straightened. In the five years since Frank had left the force, the captain had paid him exactly two visits, and both had been to warn him to stay away from active cases. Threats were all Frank had come to expect, but now it seemed the captain had a special kind of problem—perhaps the kind that would mean an end to Frank’s delinquent rent.

“I’m not on the force anymore,” said Frank airily. “Why don’t you get one of your trusted detectives to do it?”

“I need someone outside of the force,” said the captain. “Drop the act, Frank. If you don’t know what it means for me to be here, you’re not the person I need.”

Frank chuckled. “A leak? On *your* force?”

“Worse. Last night someone broke into the station’s record room and stole over 500 scrolls.”

“What were they after?” asked Frank. Without thinking, he leaned forward in his chair and reached for a fresh scroll and a quill. The movement came automatically to him, like drinking coffee or avoiding stairs.

“I don’t know,” said Donovan. “There was no pattern. They stole whole shelves of documents, everything from property disputes to expense reports. They took all the ledgers we keep on assassins, celebrities, private investigators, notaries . . . They even took both boxes of Farmer Swinson’s noise complaints. But other shelves were completely untouched. We counted at least 512 missing documents.”

“Maybe it was one of Farmer Swinson’s neighbors,” joked Frank. “They must’ve heard that after a mere hundred complaints, an intern will come to your house and give you a stern lecture.”

Captain Donovan didn’t bother to reply. He just stared pityingly until Frank cleared his throat and broke the silence. “So you want me to find these documents?”

The captain shook his head. “I want you to find the thieves. We have backups of the documents. I want to know what information they needed and what they plan to do with it.”

“A search problem,” Frank mused. During his time on the force, his two specialties had been search problems and annoying the captain.

“Does the king know?” Frank asked.

“I briefed him yesterday,” said the captain, a hint of annoyance in his voice. “Ever since the trouble with that crackpot wizard, the king insists on daily briefings on everything.” Two years ago, a megalomaniac wizard named Exponentious had tried to destroy the entire kingdom. Since then King Fredrick had personally instituted sweeping upgrades to the kingdom's security, with over 300 new security regulations, at least 5 of which dealt with the storage of official documents in government buildings under 10 stories tall.

"I can't blame him though," Donovan grumbled. "It was a close call. If it hadn't been for Princess Ann, who knows where the kingdom would be now."

Frank nodded silently. Exponentious had attacked the algorithmic foundations of the kingdom by cursing the scholars who studied those algorithms. Within months he had rendered even simple operations inefficient, and the kingdom had started to grind to a halt. Evidence of the damage had been everywhere; even in his local bakery, Frank had himself witnessed panic break out as customers discovered they couldn't remember how to arrange themselves into a line.

"The king has, of course, taken a personal interest in the matter," the captain continued irritably. "He wants all the details: Who's assigned to the case? Which search algorithms are we using? Have we scoured all of the neighboring buildings?"

Frank stifled a chuckle and mulled over the proposition. A consulting gig for the capital's police force would be good money. He glanced down at his feet, where the tip of a toe peeked through a hole in his shoe. "If I'm going to consult," he said, "I'm going to do things my way."

This was the moment of truth. Five years ago he'd been kicked off the force for *doing things his way*. The captain was a man of rules and order. Frank's last use of heuristics had been the final straw—Captain Donovan had claimed his badge that very afternoon. But, then again, doing things his own way had always gotten Frank results.

"I figured as much," the captain responded at last. He pulled a thin folder from under his trench cloak and dropped it on Frank's desk.

"I'll be in touch," Donovan said. Then, without ceremony, he turned and left the office.

Three hours and twelve mugs of coffee later, Frank sat hunched over his desk and thumbed through the thin folder of information for the seventh time. The words jumped and swayed in the flickering candlelight, but didn't provide any new insights.

There wasn't a lot to go on. The captain had given him a list of missing documents and the duty roster for the night in question, but nothing more.

Finally, with an exaggerated sigh, Frank grabbed a piece of parchment and started making notes.

The first step in any search problem is determining what it is you hope to find—the *target*, as his old instructor in Police Algorithms 101 called it. Frank had learned that lesson early; he'd been tasked in his first week as an officer with finding the duke's prize stallion, and he'd proudly returned to the station that same afternoon with a 42-pound horned turtle. Apparently, the impressive reptile wasn't good enough. A good search algorithm means nothing if you're looking for the wrong thing.

In this case it wasn't a *what*, but rather a *who*. The captain had been right about that point. Once the thieves had the documents, it didn't matter if the police got them back. The thieves already had whatever information they needed.

So his target was simple: the person or persons who stole the documents.

The second step in any search problem is identifying the *search space*. What are you searching? During Frank's daily search for his keys, the search space was every flat surface in his office. And when

Frank wanted to find a criminal, his search space was every person in the vicinity of the capital.

Frank sat back and rubbed his eyes. It was a big search problem, finding a specific criminal in a city of criminals. But he had seen worse.

Now that he had defined the problem, he could start on an algorithm. A linear search was out; he couldn't afford to question everyone in the city. He could also rule out many of the other, fancier algorithms he had studied in the academy. For a problem like this, he would have to go back to his toolkit of basic search algorithms—the private investigator's most trusted friends.

Frank made a note on the parchment. He had the target to find, he knew the search space, and he had his algorithm. It was time to get to work.

POLICE ALGORITHMS 101: SEARCH PROBLEMS

Excerpt from Professor Drecker's Lecture

In this class we'll discuss several different algorithms (and related data structures) for solving search problems. A *search problem* is defined as any problem that requires us to find a specific value (or target) within a space of possible values (a search space).

Those of you who graduate and go on to become police officers will find yourselves facing problems that fall into this category every single day. This broad definition of a search problem encompasses a lot of different computational problems, from searching the police log for a specific entry to finding rooms within a hideout to finding all arrest records that match some criteria. This class won't be exhaustive—that would take years—but I'll give you some simple examples of basic and important algorithms as we go.

The algorithms described in this class will have three common components:

Target The piece of data you're searching for. The target can be either a specific value or a criterion that signifies the successful completion of a search.

Search space The set of all possibilities to test for the target. For example, the search space could be a list of values or all the nodes in a graph. A single possibility within the search space is called a *state*.

Search algorithm The set of specific steps or instructions for conducting the search.

Some search problems will have additional requirements or complexities, which we'll touch upon as we go over different algorithms.

—2— Exhaustive Search for an Informant

The key to efficient algorithms is information." It was Professor Drecker's mantra, barked at the cadets at the start of every Police Algorithms class, ferociously enough to sear itself permanently into Frank's memory. "A good algorithm depends on finding the structure in the data and using it. It depends on information."

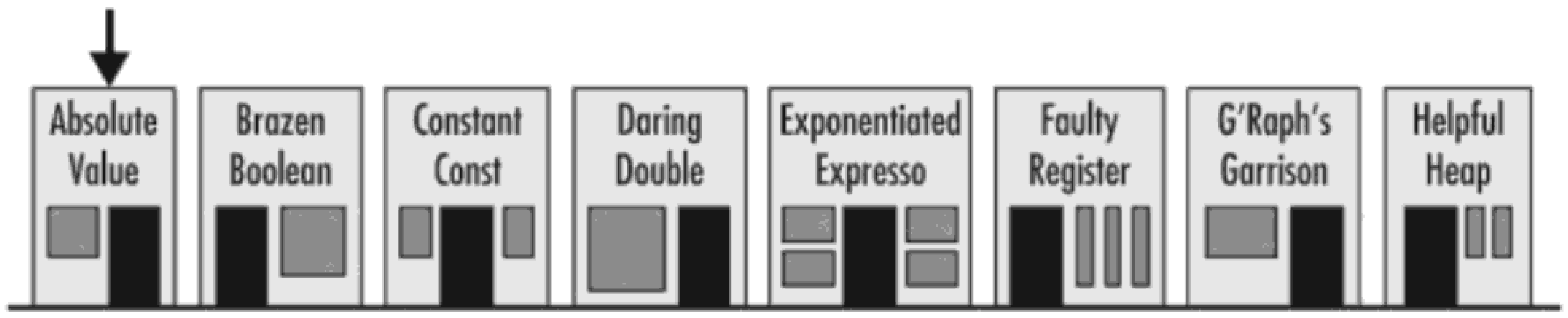
Frank smiled to himself at the memory as he turned onto Three Bit Lane, a rutted dirt road lined with a combination of seedy bars and upscale coffee shops. He nodded politely to a pair of passing knights, who clanged as they jittered past in their armor, and made a mental note to grab a Triple-Bold Espresso before he left. First he needed information, something to help guide his search. He knew exactly where to start.

Glass Box Billy would be in one of the establishments by now, sitting quietly and listening to the wisps of conversation drifting through the room. People didn't mean to say things around Billy; they simply didn't notice he was there. Billy had been blessed with a single notable talent: utter inconspicuousness. Whatever he tried, there was something about Billy that meant people just didn't notice him. Maybe it was his pale skin or his small physique; maybe it was his exceptionally mundane taste in clothing. Whatever it was, Billy had long ago decided to put his one talent to use by eavesdropping, collecting information, and selling it to anyone who would buy.

Frank eyed the eight storefronts hunched together in Three Bit Lane and wondered which one Billy would have chosen. He ran through half a dozen search algorithms in his head, but it was pointless. Frank didn't have any information to go on. Billy could be in any one of the bars or coffee shops.

He'd have to use an exhaustive search—simply try all the possibilities until he found Billy. It didn't sit well with him. Years of the detective and private investigation game had taught him that there was almost always a better algorithm than exhaustive search, and he hated resorting to something so inefficient.

Grumbling, Frank started his search. He walked into the first bar on the street, The Absolute Value.



The bartender, a surly man named Abe, glared at Frank as he entered and pointedly dropped his hand below the scarred counter. The message was clear: "I am now holding a weapon. I'll let you guess what kind. But if you hassle me, I'll give you a very close look."

"I don't want any trouble, Abe," said Frank, holding up his hands. "I'm just here to see Billy."

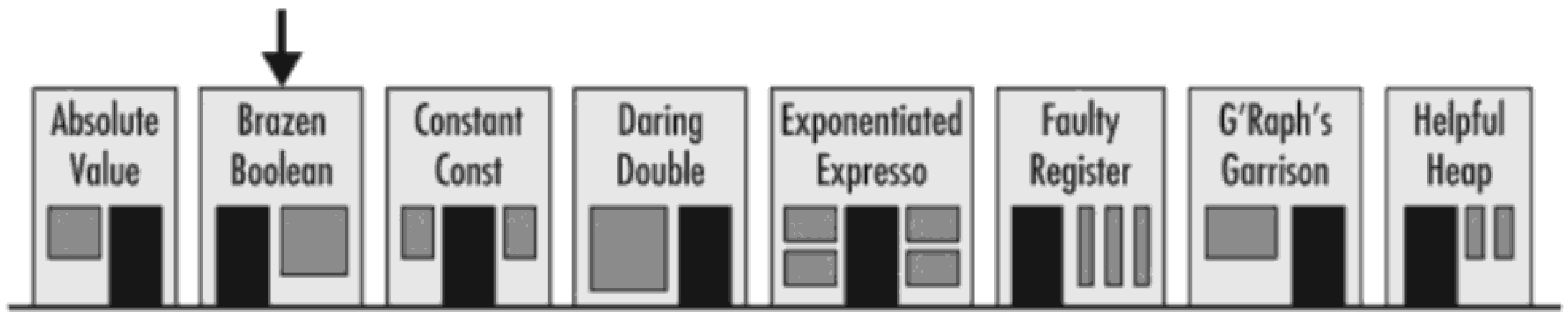
"Well, Billy ain't here," said the barman.

Frank almost smiled in relief. "Then I'll be on my way," he said.

Abe gave a curt nod and watched Frank leave, his hand still under the counter.

Frank took a couple of deep breaths and shook his head in the cool air. Abe held a grudge longer than anyone else Frank had ever met. Then again, Frank had arrested four of his siblings.

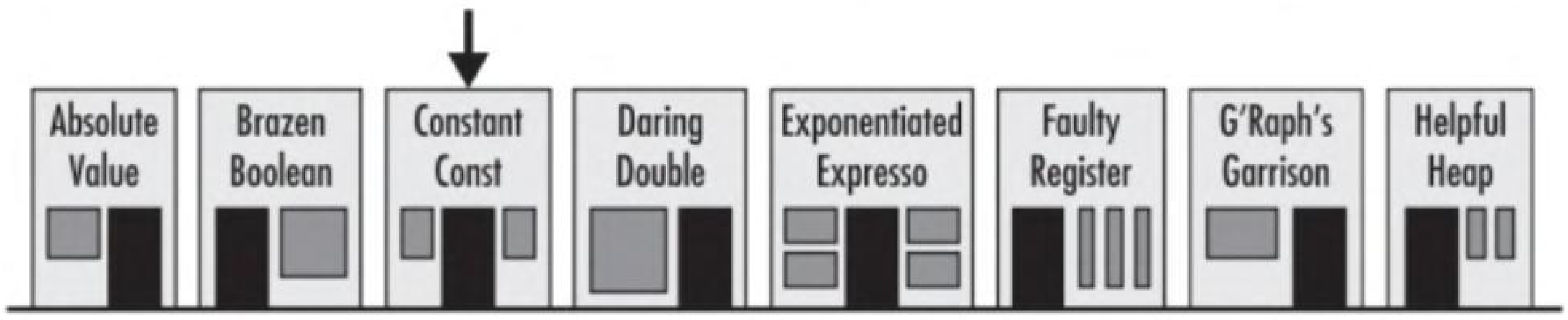
The next establishment on the street was The Brazen Boolean, a modern coffee shop decorated in typical Boolean style—stark black and white. The inhabitants of the City of Bool were renowned for their fanatic devotion to the absolute concepts of logic, viewing everything as either True or False. They made good witnesses. As the only Boolean café in town, The Brazen Boolean was a haven for expats. After all, either you were a Boolean or you weren't.



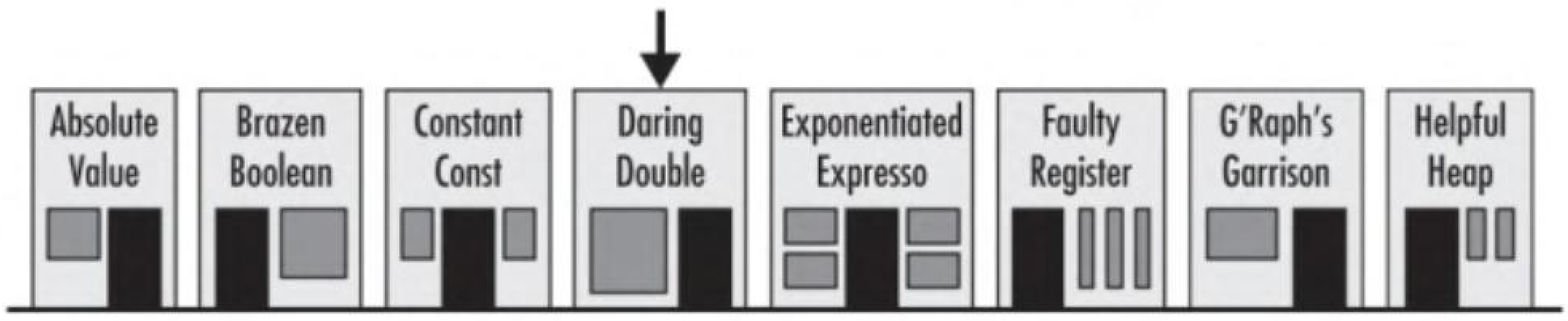
Frank popped his head in the door and asked everyone in general, "Is Billy here?" There was a brief silence as twenty pairs of eyes carefully scanned every inch of the cafe. Booleans wouldn't answer a question until they were absolutely sure.

"No," came the precise reply.

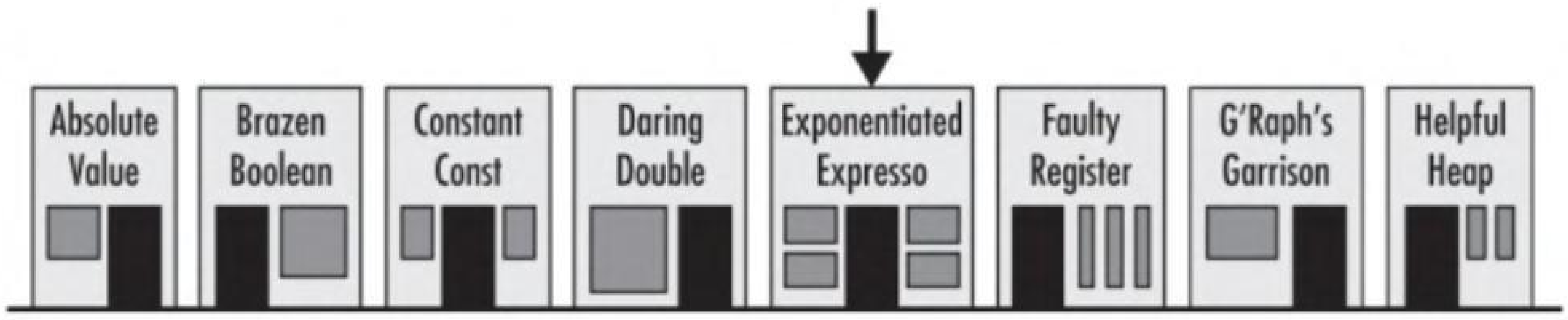
Frank continued his exhaustive search.



The third and fourth shops proved equally fruitless, although significantly more pleasant. The bartender of the Constant Const greeted Frank warmly and invited him in to reminisce about the good old days together, which was odd considering Frank had only met him the month before. And the crowd of the Daring Double, a notoriously loud wizards' hangout, cheered at each new arrival and sang happily over their steaming mugs.



Frank found Billy in the fifth shop, the Exponentiated Expresso. It was by far the loudest and tackiest coffee shop on the street, but it managed to draw the most devoted following on account of its triply caffeinated beans. On a good day, every table would be packed with jittery people who seemed to think the key to a good conversation was volume.



This morning, the Exponentiated Expresso hosted a comparatively subdued crowd. Only a handful of tables were occupied, and most of those by lone coffee drinkers who shook and mumbled quietly to themselves.

Billy sat at a central table, leaning awkwardly toward a nearby conversation. Nobody seemed to notice him. Frank had even missed him on his first scan of the room.

"Billy!" Frank called.

Billy jumped up guiltily. "Frank?" He grinned, happy that someone had acknowledged him, and sat back down. "Pull up a chair."

"I'm looking for some information," explained Frank as he took a seat across from Billy.

"Could be that I have some," said Billy. "I have such a hard time remembering these days," he said, glancing toward a long-empty mug that probably wasn't his.

Frank signaled the barista, who soon placed a fresh mug on the table. "Remember anything about a theft at the police station?" Frank asked Billy.

Billy's eyes widened and he flinched. "A robbery, you say?" he asked unconvincingly. His eyes darted around the room, but, as always, nobody paid him any attention.

Frank laid two gold pieces on the table, ignoring the sour feeling in his gut. He couldn't afford to spend this type of money, especially without knowing if he was paying for a lead or idle gossip. But he'd known this wasn't going to be cheap. He leaned in close. "Two nights ago," he said quietly, "the thieves took a whole pile of documents."

"Doesn't sound like the sort of thing that would be healthy to remember," said Billy. He eyed the gold pieces. "Afraid you're asking the wrong guy, Frank."

"That's gold," Frank growled.

"Sorry. I can't help you," Billy said. He surveyed the room again before adding, "Even if I did know something about a robbery, it's the sort of thing I would try to forget. Even if I did know something small, like who might have helped with logistics, it's not worth the risk of waking up to find my shoes packed with yak dung."

Frank stared, but Billy had gone silent. For someone who made a living sharing information, Billy had an odd habit of not saying things."Yak dung?" asked Frank.

Billy nodded, but didn't offer any more.

"You couldn't be more specific, could you?" asked Frank. "Are we talking about Northern or Southern yaks?"

"Does it matter?" asked Billy. "The point is that if I knew anything about who arranged transportation, I wouldn't remember it. Especially not if those people happened to have a large farm about five miles out of town where they could easily make someone disappear. And very doubly especially if the family that owns the farm has a history of illegal activity and an unhealthy sense of humor. Nope. It definitely wouldn't be healthy to remember anything in that case."

"Too bad," Frank said with a smile. "Maybe next time then." He nodded toward the coins. "Incentive to remember things in the future."

With that, Frank stood and strode from the Exponentiated Expresso. He turned left and continued up the street. Once off Three Bit Lane, he could swing around and make for Crannock's farm—the only farm remotely matching Billy's description.

As he passed the Faulty Register, he noticed a shadow dart into a nearby alley. He cursed under his breath, but kept going. Of course he had a tail already; the captain hadn't exactly been discreet about his visit.

But by the time he left the city and was on the rough dirt lane to Crannock's farm, he found himself in a good mood. Billy hadn't given him much, but even a little information could mean the difference between an efficient search algorithm and an exhaustive one.

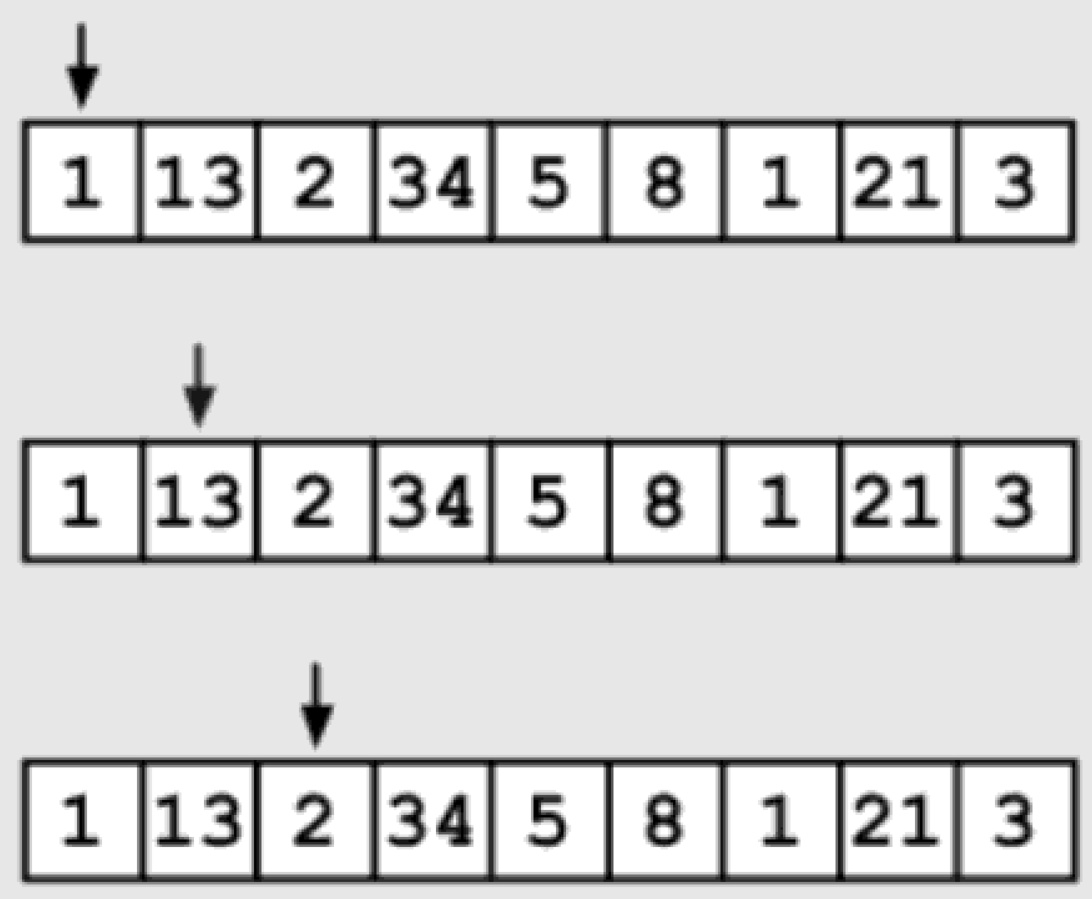
POLICE ALGORITHMS 101: EXHAUSTIVE SEARCH

Excerpt from Professor Drecker's Lecture

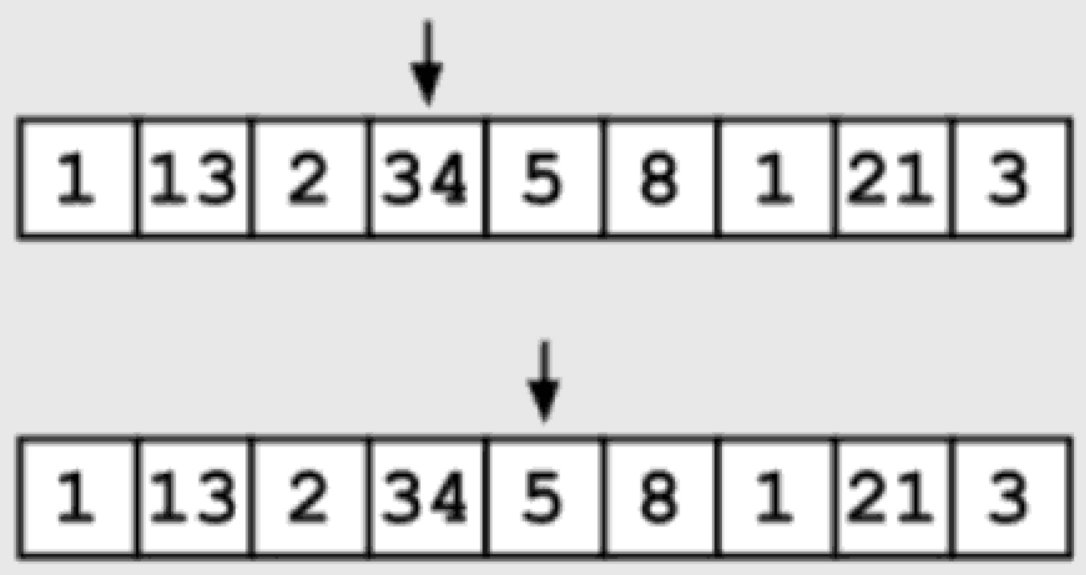
An exhaustive search algorithm searches every possibility in the entire search space for the target value. The most common exhaustive search is a *linear search*, which simply checks all the different possibilities in order.

Consider what happens when you chase a robber into the second-floor hallway of an abandoned hotel. The hall has 30 doors, all of them closed. If you've followed correct police procedures, your partner has already blocked off the opposite staircase, and the robber is trapped somewhere on that floor. How do you find him? Do you pick random doors, running back and forth until you get lucky? No! You search down the hall, kicking in one door at a time.

Or consider an algorithm that scans a list of numbers (an *array*), searching for a target value. The algorithm moves along the list from number to number, checking each value in turn so as not to miss any, and stops when it reaches the target. If we are searching an array for the number 5, then the search would progress as follows:



continued



The advantage of linear search algorithms is that they are simple to implement in the field and they work even on unstructured data. You don't have to make any assumptions about which room the robber chose; you just check everything. The downside is that exhaustive algorithms are often not the most efficient algorithm if the data has structure that can be used. If you know where the robber went, you can save yourself from kicking a lot of doors by using that information.

The key to efficient algorithms is information!

— 3 —

Arrays and Indexes on a Criminal's Farm

Frank swore aloud when he saw the police horse tied outside Crannock's house. Since the captain had gone so far as to hire Frank in person, he hadn't expected to run into any officers. If the captain didn't trust his officers, either they were under suspicion—so he'd shuffle them to some case far across the city—or they simply weren't good enough. But, from the looks of it, someone was on the case and Frank was already behind.

He slid through the open front door and joined the officer and Mr. Crannock in the foyer. Mr. Crannock shot him a disgusted look but didn't seem surprised to see him. The officer, however, seemed caught off guard.

"Who are you?" she demanded, turning on him with parchment and quill in hand.

Frank ignored her. "Mr. Crannock," he said. "So wonderful to see you again."

"Come to harass us, too?" Crannock asked. "You're not welcome here, Frank."

"I'm not looking for a welcome," replied Frank. "I'm looking for your wife. I have a few simple questions for her."

The officer stared at him. “Frank?” she asked. “Frank Runtime? Former detective turned private eye? What are you doing here? Someone lose a pet dragon?” she scoffed.

Frank ignored her again. “Your wife, Mr. Crannock. Where can I find her?”

The old man threw up his hands. “She didn’t do anything! She’s gone straight, you know. For real this time.” His acting wasn’t half-bad for an amateur.

Frank smiled; he knew its effect was unnerving. Sure enough, Crannock cringed.

“I know that, Mr. Crannock. I’m here to tap her professional knowledge. Or I could just leave the conversation to . . .”

“Officer Notation,” the young officer snapped. “And this is *my* investigation.”

That was a lie. Officers always worked these investigations in pairs. More importantly, Frank recognized Notation’s name from the duty roster the captain had given him. She had been at the station on the night of the crime.

“Officer Notation,” Frank said. “Who said I’m here on an investigation? Maybe I’m simply searching for a lost dragon.”

She scowled.

There was a commotion coming from the back of the house. Someone called for Crannock, but was cut off by a loud braying noise. “My wife is with the horses,” Mr. Crannock said impatiently. “Barn #2. Now go on, get out of my house!” Crannock waved them toward the front door and scurried away through the back.

“Thank you,” Frank called as he turned to leave. “Always a pleasure, Mr. Crannock.”

Officer Notation followed Frank across the yard. She walked hard, stomping her anger into the ground. “Do you know where you’re going?” she asked.

“Barn #2,” answered Frank.

“I know that,” seethed Notation. “But where is barn #2?”

Frank stopped and turned to her. "Just out of the academy, Notation?" he asked.

"What?"

"Only a rookie would ask a search question like that. Didn't you take Police Procedures and Data Structures? Or have they replaced that course with something less rigorous—Introduction to Turtle Graphics, perhaps?"

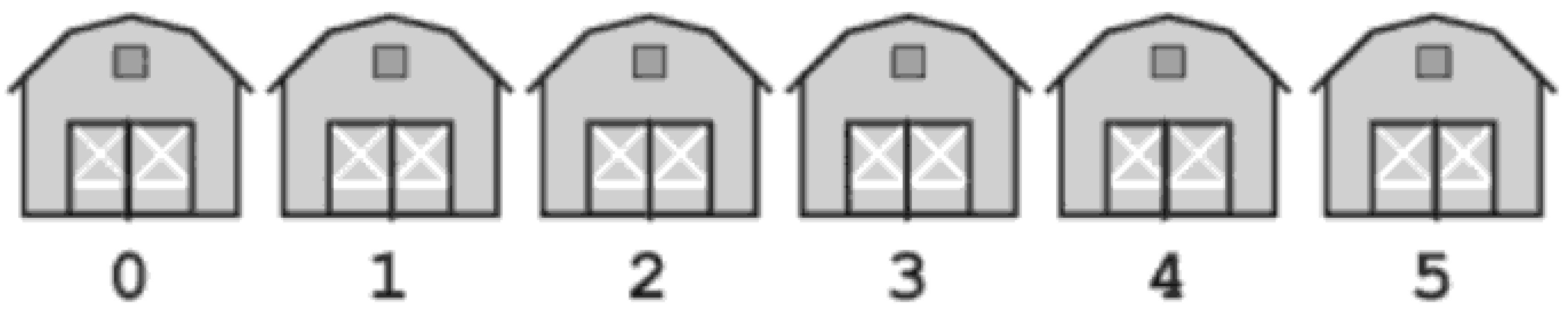
Notation seemed taken aback. "Of course I took Police Procedures and Data Structures," she said, though she sounded uncertain. "But what I meant was—"

Frank cut her off, "Then you know about arrays and indexes."

"Yes, but—" started Notation.

"Finding a barn on a farm is a simple enough search task," Frank interrupted again. "We could use an exhaustive search to check each building. *FOR EACH building on the farm: check if it is barn #2.* Back in my day, you learned that search on the first day of Police Algorithms.

"But we can do better here. The Crannocks have six barns in a nice line—just like a giant array. Mr. Crannock was kind enough to supply us with the barn number, the index into that array. All we have to do is walk to the corresponding barn."



"That's not what I meant!" shouted Notation, waving her arms. "I know how to use the index of an array. I know that we only have to walk up to the barn with a giant #2 outside. I graduated first in my class in both Data Structures *and* Police Algorithms, so don't lecture me on the correct use of arrays."

"Well, you asked," Frank replied.

"What I was asking is: Do you know where this wonderful array of barns is located?"

"Of course you were," said Frank. He began walking again. "You still sound like a rookie, though, quoting class rank."

"Where are the barns?" shouted the officer, stamping to catch up.

Frank shot her a smile over his shoulder. "Over this hill."

As Frank had learned years ago, the Crannock family embraced the concept of arrays with an almost fanatical devotion. They organized everything into linear structures with clearly labeled indexes for each element. As he passed barn #0, Frank noted 15 pig troughs, each capable of storing one serving of food. A farm hand was iterating down the line and ladling out the next meal into each array location.

Frank and Officer Notation moved on to barn #2, labeled with a sign outside its door. Mrs. Crannock's icy greeting was almost pleasant, compared to previous encounters; she hadn't even thrown anything . . . yet.

"What do you want?" Mrs. Crannock demanded.

"Mrs. Crannock," Notation cut in before Frank could steal her witness. "I was hoping I could ask you a few questions."

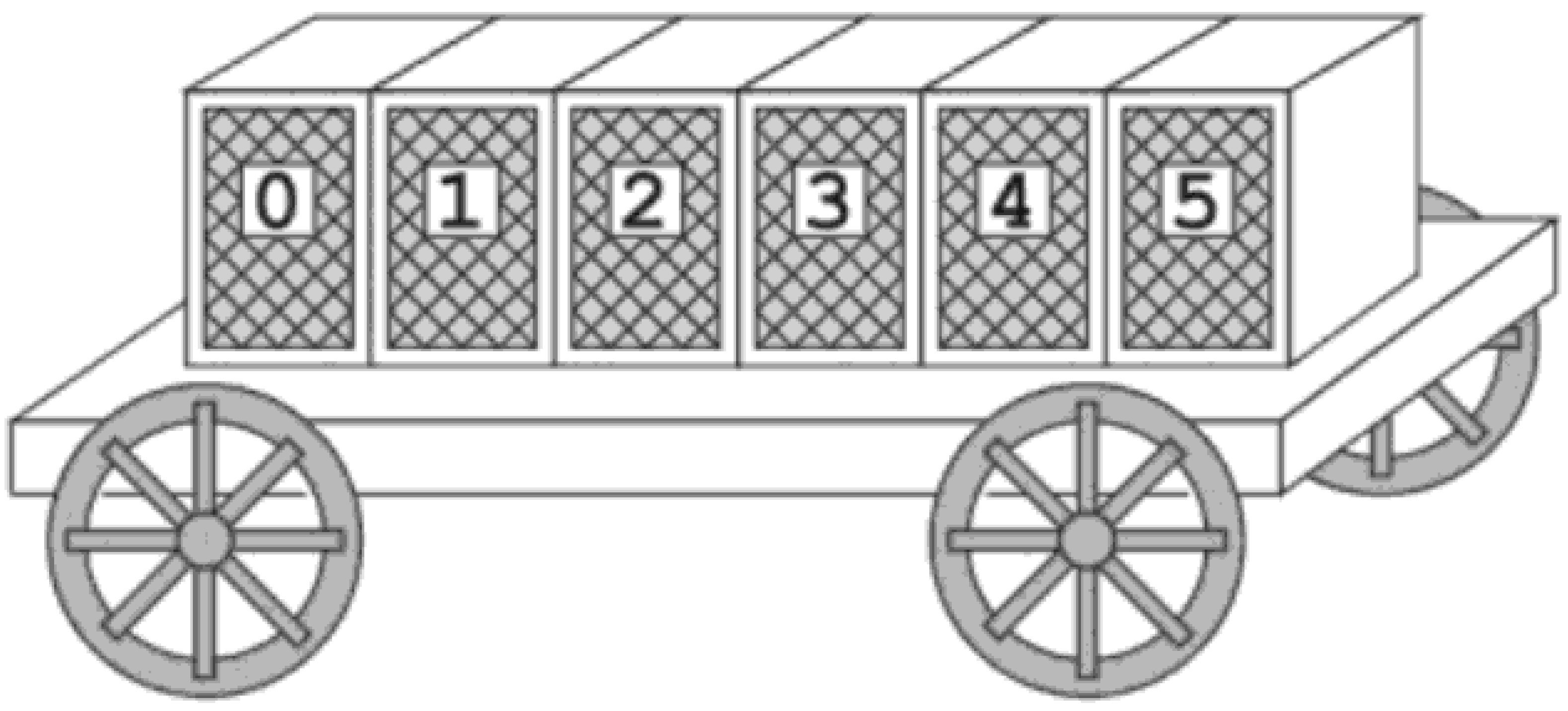
Frank let Officer Notation ask the questions. Billy's clue hadn't yielded anything more than a lead to the farm, but Notation appeared to be working from a better collection of clues.

Mrs. Crannock sneered and spat on the ground. "I didn't do anything," she said. "I've gone straight, you know."

"I'm not here to arrest you," said Notation. "I need to ask you about a certain donkey cart—the ArrayCart?"

A flicker of doubt went through Frank. Could Officer Notation be here for a different case? He doubted it. His gut told him that she was after the lost documents, and he had learned to trust his gut.

"The ArrayCart," said Mrs. Crannock suspiciously, though with the barest note of pride. "My own invention. Based it off of an array. It's got individual storage pens for our animals. Each pen stores exactly one animal. Since they all have separate doors, you can walk up to any pen and take an animal out or put one in. Easy access to any storage location. Saves hours of wrangling."



"It's quite ingenious," Officer Notation conceded. "You've found a way to apply the concept of arrays and indexes to livestock transportation."

"And that's just the beginning," added Mrs. Crannock. "I'm working with a certain *wizard* on a completely new type of ArrayCart—one with magical pointers! I bet they'd be perfect for the police force. Tell your captain that I can give him a good price."

Frank had to hand it to Notation. The surest way to get a Crannock talking was to bring up arrays.

"You have a few ArrayCarts that you rent out now. Is that correct?" probed Officer Notation.

Mrs. Crannock's eyes became instantly cold. "It's a legitimate business. We pay our taxes."

Frank held back a derisive snort.

"Did you happen to rent an ArrayCart to anyone two nights ago?" pressed Officer Notation. "A smaller model with six pens."

"I might've," said Mrs. Crannock. Her cold demeanor was creeping toward hostile.

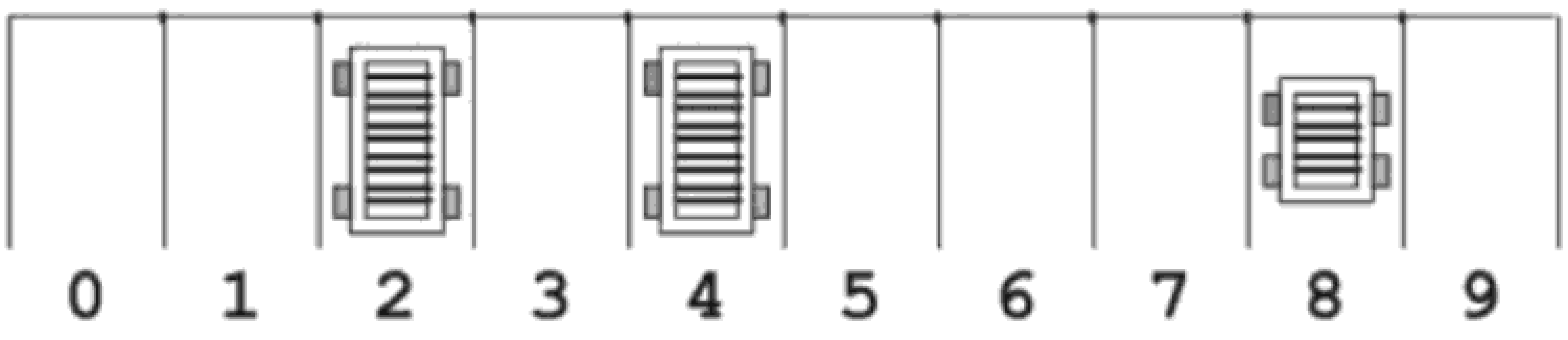
"Do you have a record of who rented it?" asked the officer.

"No," said Mrs. Crannock. "We shred the records once the carts are returned. I don't happen to recall who rented that one."

It seemed like Billy's hint had paid off. If you were a criminal in need of transportation, there were few places that would rent you a cart, and fewer still that would forget your name afterward. Mrs. Crannock may have claimed to have gone straight, but apparently she still, at the very least, provided a valuable service to her former associates.

"Are you sure you don't remember anything about your customer?" prompted Officer Notation, but Frank knew it was pointless. He had once questioned her for three hours about a stolen yak. She hadn't given him a single peep, despite being the one who had been robbed. Mrs. Crannock wouldn't talk.

While Officer Notation tried a few variations of the same question, Frank quietly slipped out of the barn and found the cart lot. As he expected, the lot was organized as an array with 10 labeled parking spots. Only spots #2, #4, and #8 were occupied. The carts in positions #2 and #4 had 10 pens apiece, so were too large to fit Notation's description. But slot #8 held a six-pen ArrayCart, its wheels still coated with fresh mud.



After a quick glance around, Frank heaved himself into the back of the six-pen ArrayCart. A scattering of straw covered the floor, but the cart was otherwise empty. Frank opened each pen in turn, scanning the empty storage spaces for any clues. Then, getting down on his hands and knees, he sifted through the straw until he found a few scraps of parchment.

He collected six tiny pieces in all, probably corners that had caught on nails as the scrolls were unloaded. Only two of the pieces contained writing, and those appeared to be from ledgers. It wasn't exactly a solid lead, but it tied the cart to the crime.

Frank moved his search to the front of the cart, carefully inspecting everything around the driver's seat. The seat itself gave him his first real clue. There he found a few black and orange threads caught by the seat's splintered wood. From the vividness of the colors alone, Frank could tell the cloak must have been new. Satisfied, he pocketed the threads and stepped down from the cart.

Only when a gust of fresh air hit him did he notice he'd been holding his breath. Around the cart was a stench of rotting fish. He sniffed lightly, following the smell, and arrived at the mud-caked wheels. He took a great, deep sniff and immediately regretted it. The smell of rotting eel emanating from the mud was as unmistakable as it was unpleasant.

Frank half-smiled, half-gagged as he staggered back from the cart. He might not know who had rented it, but now he knew where it had been.

POLICE ALGORITHMS 101: ARRAYS

Excerpt from Professor Drecker's Lecture

Arrays are simple data structures that allow you to store multiple values. An array is like a row of bins. Each bin can store a single piece of information, such as a number or a character.

Value:	20	15	19	1	10	1	5	33	9
Index:	0	1	2	3	4	5	6	7	8

The structure of an array means you can access any value (or element) within the array, whether to write to it or read from it, by specifying its location, or *index*, within the array. Many programming languages use 0-indexed arrays, which means the first value of the array resides at index 0, the second at index 1, and so forth. Commonly, you reference the value at index i of array A as $A[i]$; for example, the third element of array A would be $A[2]$ and equal to 19.

You likely recognize this structure from your introductory tour of the holding cells in the capital's police station yesterday. The king personally suggested the use of indexed, single-person cells to streamline the retrieval of prisoners. Each station is equipped with an array of four to eight holding cells, depending on the size of the local criminal population.

—4—

Strings and Hidden Messages

Frank shook off Officer Notation and left through the back gate of the farm, where a large sign faced the road. For years the Crannocks had used this sign to broadcast coded messages about various illegal activities. These days it was something of a tourist attraction for visiting criminals—a place thugs took their younger protégés and gathered to reminisce about stories that invariably began "Back in my day . . . "

The sign itself was an AnyText model. It held 3 arrays of letters, each array with 12 slots. Each letter, space, or punctuation mark took up a single slot in an array, meaning the board could hold a total of 36 individual characters—enough to advertise a whole range of illegal activities. Every Monday morning, one of the Crannocks would drag a basket of letters to the sign and individually place the appropriate character in each slot of the array.

During his first week on the force, Frank's partner had brought him out here to "check the board." The message at the time—*Apple picker wanted. Got slugs?*—sounded innocuous enough to Frank. The Crannocks were looking for an apple picker to help with the harvest and were offering to get rid of people's slugs. When he said this to his partner, a 20-year veteran, she laughed.

A	P	P	L	E		P	I	C	K	E	R
W	A	N	T	E	D	.					
G	O	T		S	L	U	G	S	?		

“That’s what they want you to think,” Detective Rossile explained. “You have to look beyond the obvious meaning and see what the criminal mind would see. In this case, *Apple picker wanted* indicates that they are trying to hire a petty thief. Someone who would steal apples from a cart or such.”

“And the slugs?” Frank asked.

“Illegal slug racing,” she replied. “They hold races here every few months. You’ll get to know that one.”

Thus, Frank had learned to check the Crannocks’ board weekly to get a pulse on the criminal world. After the first few months, he had learned to decipher most of the codes. *Farmhands* meant henchmen, with additional modifiers if strength, brutality, or just plain numbers were needed. A *print artist* referred to a forger, while a *vocal artist* was a con man, and so forth. The phrase *a flock of chickens* had stumped Frank for a few days before Rossile translated it as “a large number of warm bodies to run around noisily and cause a distraction; no intelligence required.”

By the end of his first year, Frank had become an expert at reading the board. The only time in the past few years that Frank had a hard time deciphering a criminal tip from the board was during the wizard Exponentious’s attack on the kingdom. Exponentious had unleashed the Spell of Incorrect Indexes on all the kingdom’s ArrayDesignBoards. As its name implied, the spell changed the indexes, so the locations Mrs. Crannock thought she was setting the letters to were wrong. For a week, the Crannocks’ board held gibberish.

D	E	F		V	E	I	N	S			E
D	I	Z					W	A	R		
D	E	N	T						A	W	

Since the spell only mixed up letters within an array and the AnyText model was implemented as three separate arrays, Frank had to unscramble each line individually. He puzzled out the message: *Defensive wizard wanted.*

Today, though, the message was clear. In fact, it was the least subtle message he had ever seen on the Crannocks' board. It read *ArrayCarts for rent. No questions.*

ARRAYCARTS
FOR RENT.
NO QUESTIONS

POLICE ALGORITHMS 101: STRINGS

Excerpts from Professor Drecker's Lecture

Arrays don't just store lists of numbers; they can also be used to store strings of text characters. Many programming languages implement strings using arrays. Each block in the array holds a single character, which can be a letter, number, symbol, or space. As with arrays of other data, characters in these strings can be accessed directly through their index in the array.

Value:	H	E	L	L	O	!
Index:	0	1	2	3	4	5

During your career in the police force, you will come to know this representation of text *very* well. All standard police forms require officers to record their names within a 32-block array at the top of *each* page. In a typical month, you will fill in over 400 such arrays.

— 5 —
Binary Search for a Smuggler's Ship

The port of Usb was little more than a fishing village. A dozen weathered buildings clustered around the end of a single long pier. A few pockets of meager activity surrounded the most recent arrivals, but otherwise the town was reassuringly quiet.

Frank headed straight for the Crab's Pinch, a fisherman's bar renowned for its clam chowder and Wednesday night sea shanty contests. With any luck, one of his contacts would turn up before the day was out. After all, the Crab's Pinch was the only place to go in Usb. So Frank planted himself at a table in the back corner, ordered the chowder, and waited.

It wasn't long before a freelance smuggler named Mavis entered the dank little bar. Careful by nature, Mavis had never technically been convicted of a crime, though it was well-known that she'd once set her own ship on fire to destroy evidence. Frank got along with her well enough, at least once he'd left the force, and they even exchanged the occasional scrap of information.

Frank, having nursed his chowder for a solid hour, finally pushed away his bowl and motioned to Mavis. She hesitated a moment by the door before jostling her way through the bar.

"Mavis," said Frank as she joined him in the corner, "how are you?"

"I was doing a lot better 10 minutes ago," she spat.

Before Frank could ask, Officer Notation strode through the door and held up her hands. "Ladies and gentlemen," she called. "If I could have your attention for a moment. I'm looking for a cart that came through here two nights ago."

Frank cursed under his breath. So much for his lead.

"I come in from the dawn run, hoping for a bowl of hot chowder and a few minutes of peace," Mavis complained. "Instead I get this copper clammering about donkey carts."

Frank laughed dryly. "And until she goes away, you can't unload your cargo. Right?"

Mavis scowled at him but didn't object. Usb had never found success in either the fishing or shipping industries. The port did, however, appeal to those criminals concerned with moving merchandise without dealing with nosy government officials. Frank would wager a month's rent that there wasn't a single ship at dock that wasn't smuggling something.

"Do you know anything about the cart?" Frank dropped his voice to just above a whisper.

Mavis shrugged. "There's always carts on the docks. This is a port, Frank. People move things."

"This is a special cart," Frank pressed. "A bunch of individual animal pens, like a giant array on wheels."

"Sounds fancy," said Mavis. "But I haven't heard of any ships moving animals. I might have heard a rumor about a crate or two of miniature turtles, but nothing large enough to need a pen. You sure it came through here?"

Frank nodded. The smell had been like a fish-scented air freshener in an outhouse, and few places smelled as bad as Usb.

"Anybody casting off at that time?" he asked. If the thieves had transported the stolen documents this far, they wouldn't have waited around.

"Only the *Retry Loop*," said Mavis. "And I'm only telling you that because it's public knowledge. I don't know what it was carrying, and I don't care."

"Do you know when it returns?" asked Frank.

"Got back into port 19 hours ago," replied Mavis. "Don't know what it was carrying then either."

Frank smiled widely. "Sounds like it's time for me to take a stroll around town," he said.

Mavis smiled halfheartedly at him and turned to flag down a waiter.

Frank made it less than 20 meters down the pier before Officer Notation marched up beside him.

"Mr. Runtime, this is my investigation," she began. "If you have information—"

Frank stopped, causing her to pull up short. “What exactly are you investigating, Officer?” he asked.

It was better than Frank had hoped. Notation opened and closed her mouth a few times as a red flush spread up her neck.

“The captain doesn’t know you’re here, does he?” Frank asked. “This isn’t exactly an *official* investigation.”

“I don’t know what you’re—” started Officer Notation, but Frank cut her off.

“Cut the act,” he said. “The fact you’re out here alone is all the proof I need. You’re running this investigation on your own time. The question is, why?”

The flush had now finished its ascent of Officer Notation’s face. Her ears burned a particularly vivid shade of red.

“That’s none of your concern,” she said.

“It is when the captain comes to me because he can’t trust his own officers,” Frank replied calmly.

“The *captain* hired a washed-up gumshoe like *you*?”

“Yes. Because he can trust *me*.”

Officer Notation’s face grew hard and her eyes burned. For a second, Frank thought she might end this conversation with her billy club. But almost as quickly as her anger had flared, it deflated.

“I need to recover those documents,” she said mournfully. “It was my fault—I was on guard duty that night.”

“I see,” said Frank thoughtfully.

“I need to recover those documents,” repeated Officer Notation, sounding agitated. “I’ve only been on the force for a few months and—”

Frank cut her off and gave her what he hoped was a reassuring smile. This was what he had expected. Rookies rarely dealt well with their first mistakes, and Notation seemed more tightly wound than most. “We’re looking for the *Retry Loop*,” he said. “The Crannocks’ cart unloaded something there the night of the robbery. The ship docked 19 hours ago.”

He didn't trust her, of course, but he wanted to keep her close, keep an eye on her. The fact that she had found the Crannocks meant she knew more than she had put in her report. Something was missing from her story, and he needed to find out what else she knew.

"We better get started," said Notation, looking worriedly down the pier. "There are a lot of ships to check. Should we start at the front?" As most of the vessels in port belonged to smugglers, none of them displayed identification. They would have to ask each ship's name in turn.

"We can do better than that," Frank explained. "The harbormaster is fanatical about organization. He insists that the docked ships be sorted in order of their arrival time. The newest arrival gets a prime spot near town, where the crew can easily load and unload, but when a new ship arrives, the rest of them are forced to shift down to give it space in front."

"That's absurd," protested Notation. "What a tremendous amount of wasted effort. Why would he do that?"

Frank chuckled. "He claims it's for efficiency, but anyone who's spent a week in Usb knows the truth. The harbormaster can't stand the smell of rotting fish. Ships that remain in harbor without selling their loads become, well . . . fragrant. The harbormaster's organizational scheme moves the ones that have been here longer away from his shack."

Officer Notation stared at him. "Are you serious?" she asked finally.

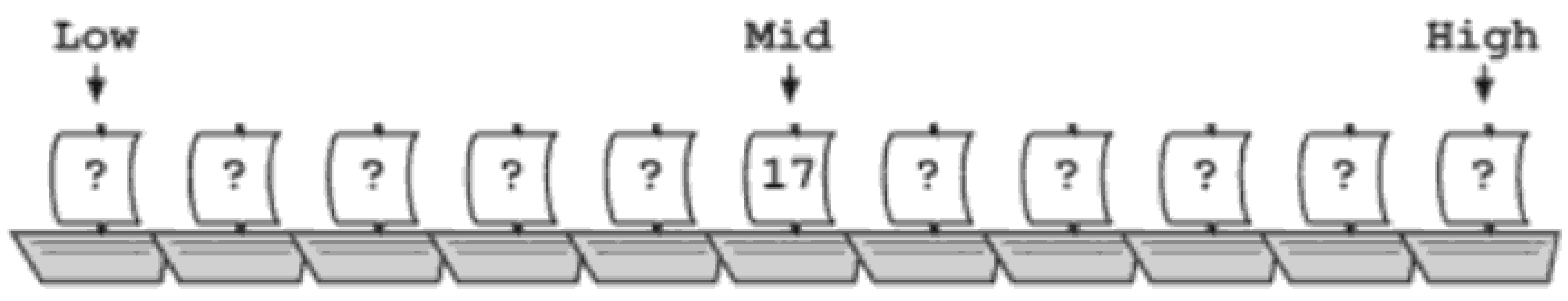
Frank chuckled again. "Yes. You'll start picking up these useful bits of information, too, once you've walked the beat awhile. The point is that we know the ships are in sorted order and we know the *Retry Loop* has been here for 19 hours, so we can just do a *binary search*.

"Our target value is 19, and our algorithm is binary search. Right now the search space is that whole line of ships, so we already have an upper and lower bound. If we use inclusive bounds, our lower bound is the first ship and our upper bound is the last ship. If the *Retry Loop* is here, it obviously can't be in front of the first ship or after the last ship.

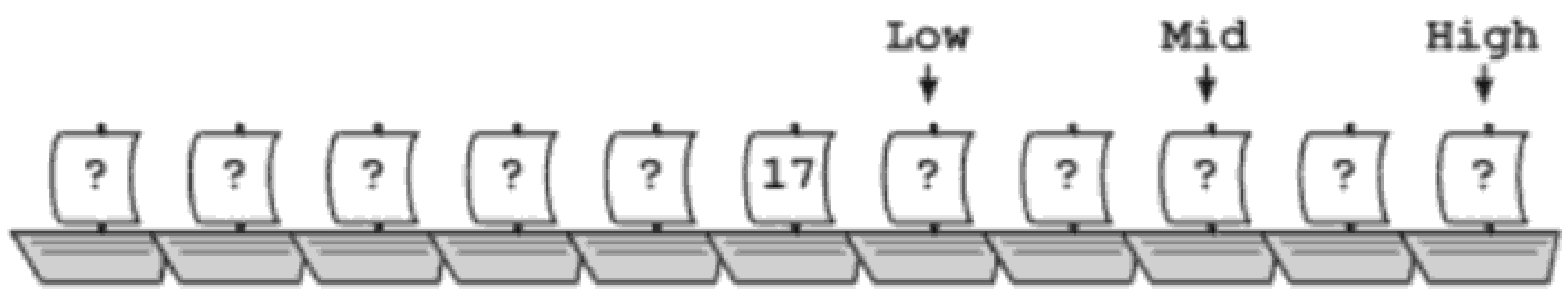
"So we start with the middle ship and ask how long it's been in port. If it's been there less than 19 hours, then it must come before the *Retry Loop.* That will split our search space in two. And—"

"If it's been there more than 19 hours, then it must come after the *Retry Loop,*" interrupted Notation. "I know about binary search. My Police Algorithms final was just two and a half months ago."

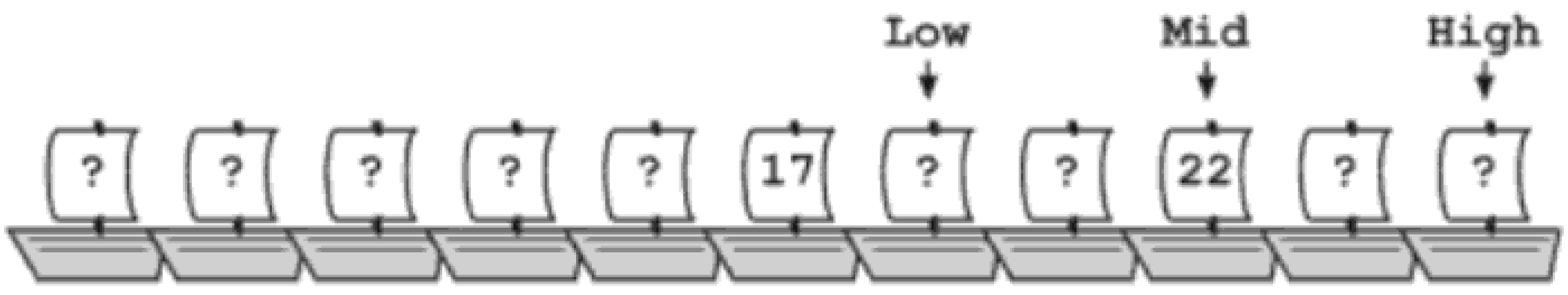
With that, the two of them set off in search of the *Retry Loop.* The middle ship, a yellow schooner that smelled oddly of bananas, had been in port for 17 hours.



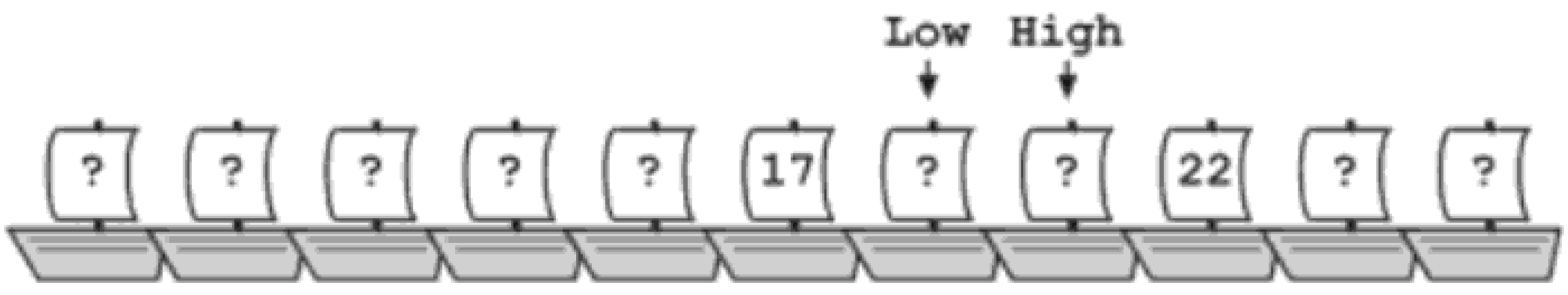
That meant they could rule out the half of the ships at the front, including the middle ship. Frank adjusted the lower bound to the first ship that *could* be the *Retry Loop,* one ship past the yellow schooner.



With the reduced search space, they chose a new middle point. It took a while to convince the captain of the next ship that they weren't undercover customs officials. After 10 minutes, Notation shoved her badge under the captain's nose, and his tone changed immediately to an irate whine as he informed them that his ship, the *Corrupt Packet*, had been stuck in port for 22 agonizing hours. He demanded they speak to the harbormaster on his behalf.

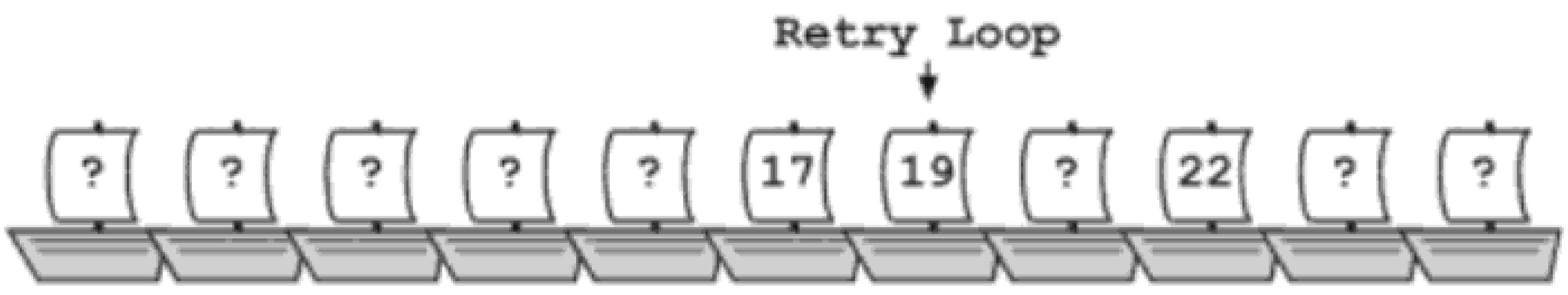


Since their target was 19 hours, they knew the *Retry Loop* would have to come before the *Corrupt Packet*. They changed the bounds again so that the ship to the left of the *Corrupt Packet* was now the upper bound.



This left only two ships in the search range; they were rapidly nearing the end of the search. If neither of these ships was the *Retry Loop*, they would know for certain that it had left port, as once there were no more elements in the search space, they could rule out the entire search space.

Since there were only two ships left, they could choose either as their new middle point. Going with his gut, Frank picked the earlier ship, which happened to also be their lower bound. A quick chat with a crewmember loitering on the pier confirmed that the ship was indeed the *Retry Loop* and it had been in port for 19 hours.



"Now what?" asked Officer Notation as they stood watching the ship.

"We use your shiny badge again," Frank replied.

POLICE ALGORITHMS 101: BINARY SEARCH

Excerpts from Professor Drecker's Lecture

A binary search algorithm is used to efficiently find a target value v in a sorted array A. Unlike in a linear scan, a binary search uses information about the structure of the data to make the search more efficient. The key to efficient algorithms is information. In this case, we use the fact that the array is sorted in increasing order:

$A[i] \leq A[j]$ for any pair of indexes i and j such that $i < j$

This might not seem like a lot of information, but it's enough to make the search more efficient.

The binary search algorithm works by repeatedly dividing the search space in half and limiting the search to only one of those halves. The algorithm limits the active search space by tracking two bounds. The upper bound (*IndexHigh*) marks the highest index of the array that's part of the active search space. The lower bound (*IndexLow*) marks the lowest index. Throughout the algorithm, if the target value is in the array, we guarantee the following:

$$A[IndexLow] \leq v \leq A[IndexHigh]$$

At each step in the search, we check the value halfway between the lower and upper bounds:

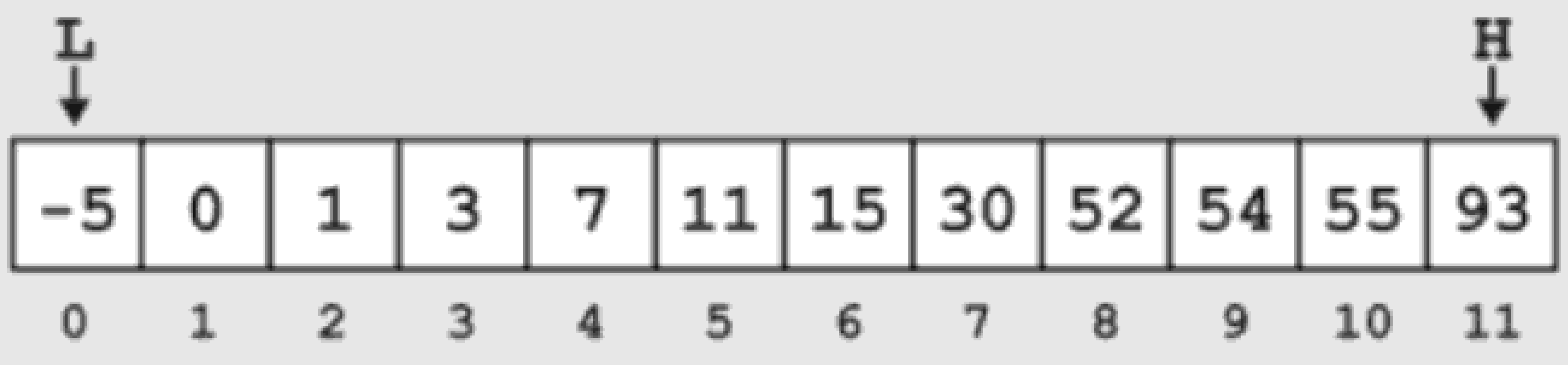
$$IndexMid = \frac{IndexHigh + IndexLow}{2}$$

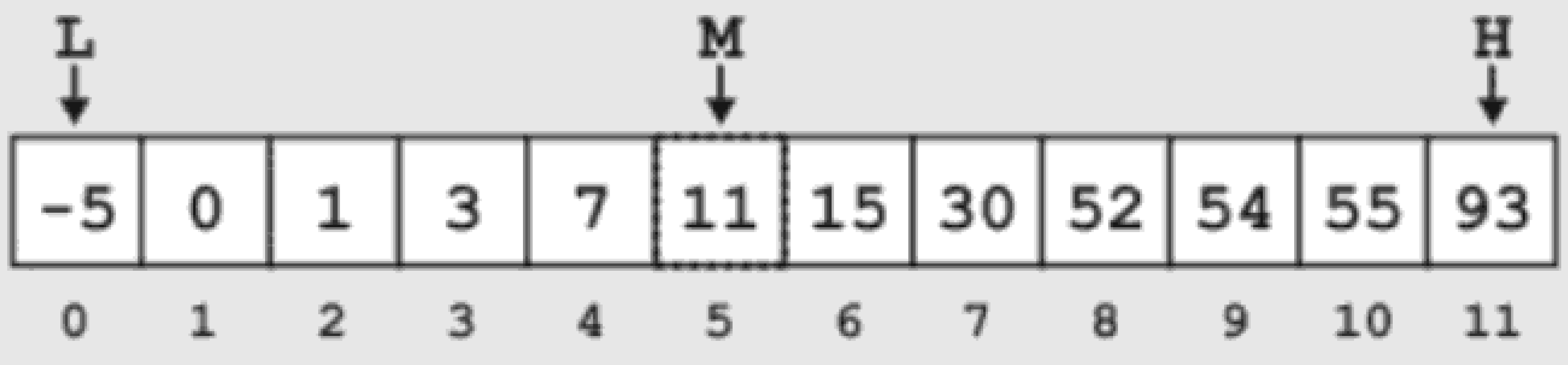
We can then compare the value at this middle location, $A[IndexMid]$, with the target value, v. If the middle point is less than the target value, $A[IndexMid] < v$, we know that the target value must lie after the middle index. This allows us to chop the search space in half again by making *IndexLow* = *IndexMid* + 1.

If the middle point is greater than the target value, *A*[*IndexMid*] > *v*, we know the target value must lie before the middle index, which allows us to chop the search space in half by making *IndexHigh* = *IndexMid* – 1.

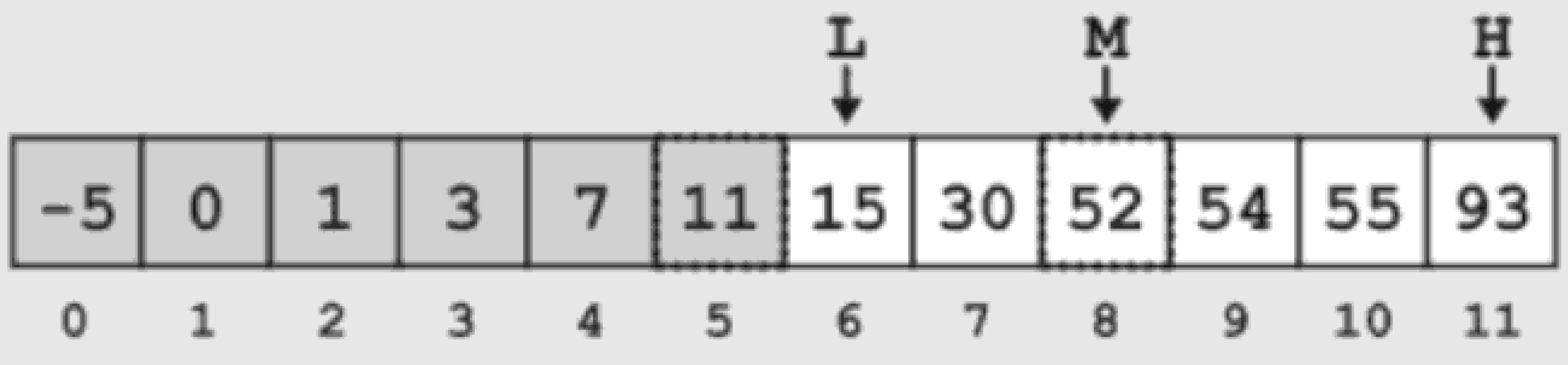
Of course, if we find *A*[*IndexMid*] equals *v*, we can immediately conclude the search. We found the target.

Let's consider searching the following (sorted) array for the value 15. The boxes with dotted outlines correspond to the values the algorithm has checked, and the shaded elements are ones that have been eliminated from the search.



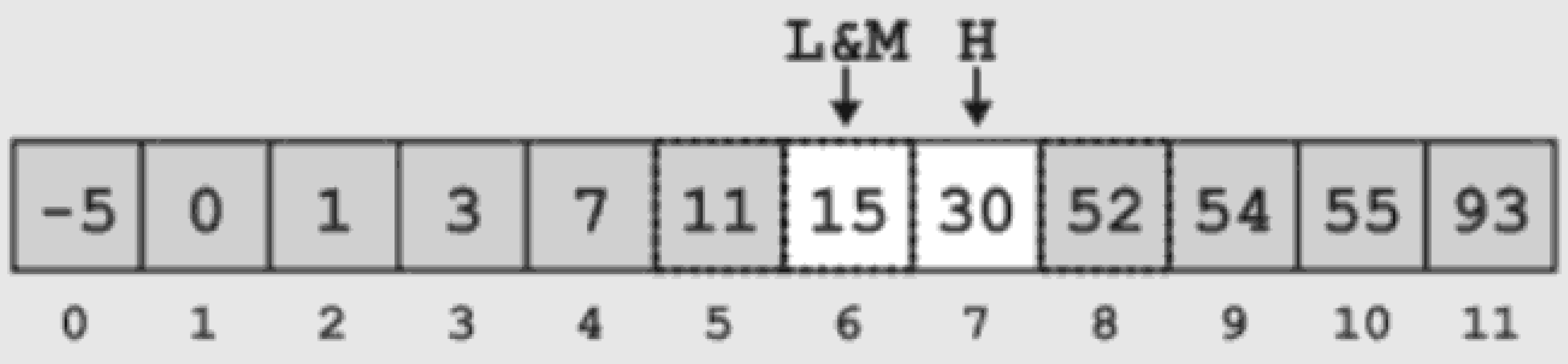


The first midpoint check finds a value of 11, which is less than our target value of 15. Since we know the array is sorted in increasing order, we can rule out the midpoint and anything before it. We move our lower bound index appropriately (*IndexLow* = *IndexMid* + 1).



continued

Similarly, after the second comparison, we find a midpoint value of 52, which is greater than the target value. We can rule out the midpoint and everything after it. We move our upper bound index (*IndexHigh* = *IndexMid* – 1).



Note that even though the lower bound's index pointed to the target value (*v* = 15) for several iterations, we continued the search until the midpoint pointed to the target value. This is because our search checks only the value at the midpoint. We don't check the values at the lower or upper indexes until the midpoint reaches them.

What happens if the target value is not in the array? As the search progresses, the bounds will move closer until there are no unexplored values between them. Since we are always moving one of the bounds *past* the midpoint index, we can stop the search when *IndexHigh* < *IndexLow*. At that point we can guarantee the target value is not in the array.

— 6 —

Binary Search for Clues

Food inspectors," Frank called out as he and Officer Notation stepped up the narrow gangplank and onto the ship. At Frank's instigation, Notation waved her badge in a blur, too fast for anyone to read.

"Food inspectors?" asked a crew member. "We aren't transporting any food."

Frank looked the man over. He wasn't an officer or hired security, probably just a sailor who had taken charge while the officers were away. It wasn't uncommon. Smugglers rarely employed guards to watch their ships. It drew too much attention.

Frank turned on the sailor, growling out his words. "We'll see about that. I'm told there's a load of rotting eels on this dock, and I intend to find them."

"Eels?" The sailor was clearly out of his depth.

"*Rotting* eels," Frank shot back. "We're going down to check the stores." Then, without waiting for a response, he strode to the hatch leading below deck.

Notation hurried after him.

"We don't have much time before they get the captain. We need to find the logbook," Frank said as he climbed down the ladder. The logbook would contain a manifest of items shipped and a list of ports visited. The manifest would be fake, of course. Smuggling ships never documented their true cargo. But with any luck, he could read between the lies and find a clue.

Officer Notation found the logbook at the back of the hold and pulled it out. Frank checked the cover and swore:

> Manifest and Log of the *Retry Loop*
> Captain: A. James
> Home Port: Usb
> Owner: Vinettees Shipping Group LLC

After months of successfully avoiding the Vinettees, Frank had walked right onto one of their ships. He found himself reflexively scanning the hold for hidden henchmen, stashes of weaponized farm equipment, or evidence of a slug racetrack. Frank discarded the last possibility—everyone knew slugs wouldn't race on a ship; it had something to do with being surrounded by large quantities of saltwater.

He shook his head and focused on the problem at hand. Frank had to find a clue before the Vinettees knew he was on the ship, or he might not get back off again. He turned to the end of the book and started flipping toward the front, one page at a time.

"What are you doing?" asked Notation.

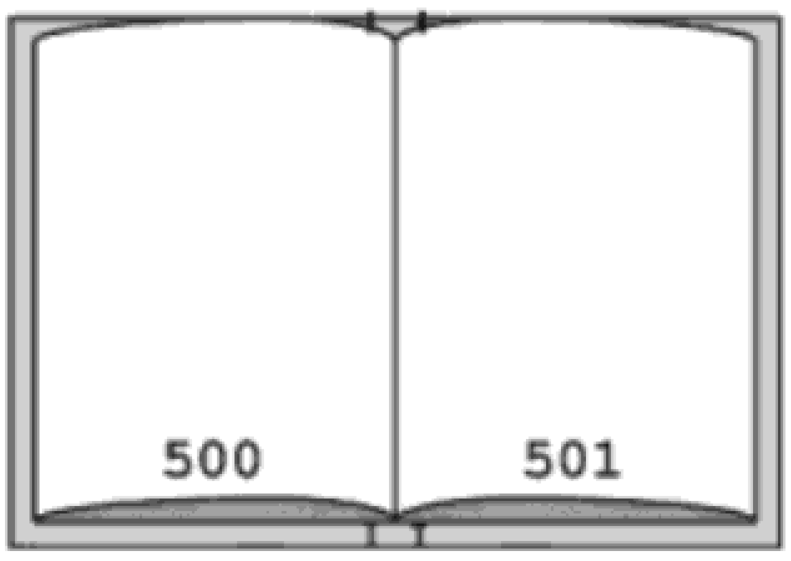
"Looking for the last entry," said Frank.

"One page at a time?" asked Notation. "There's got to be a thousand pages. Why don't you use binary search again? We just used it two minutes ago."

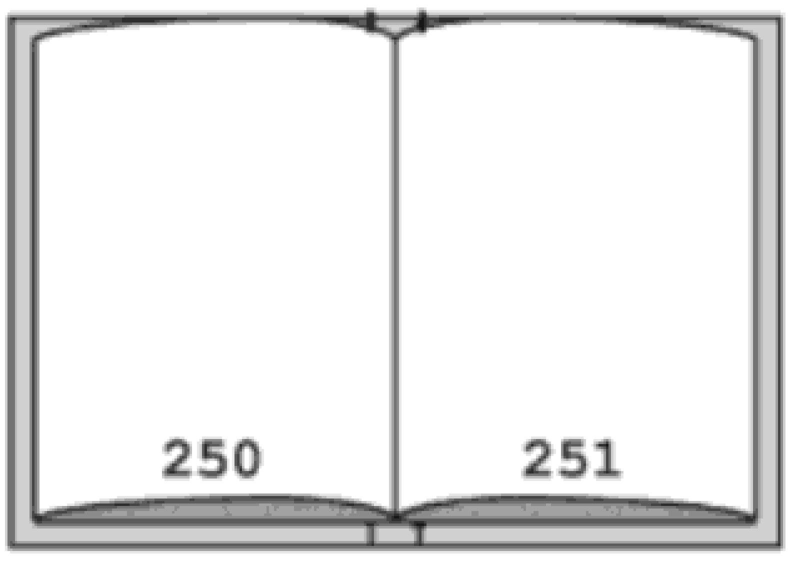
Frank paused. He wasn't looking for a specific page number, but he could still use a binary search to find the last entry. He would just refine the search bounds depending on whether the current page had text or not.

"Okay. Binary search," he agreed.

He opened to the last page again and confirmed the book had exactly 1,000 pages, giving him a lower bound of page 1 and an upper bound of page 1,000. He added the numbers, divided by 2, and computed a midpoint of 500. He flipped to that page.

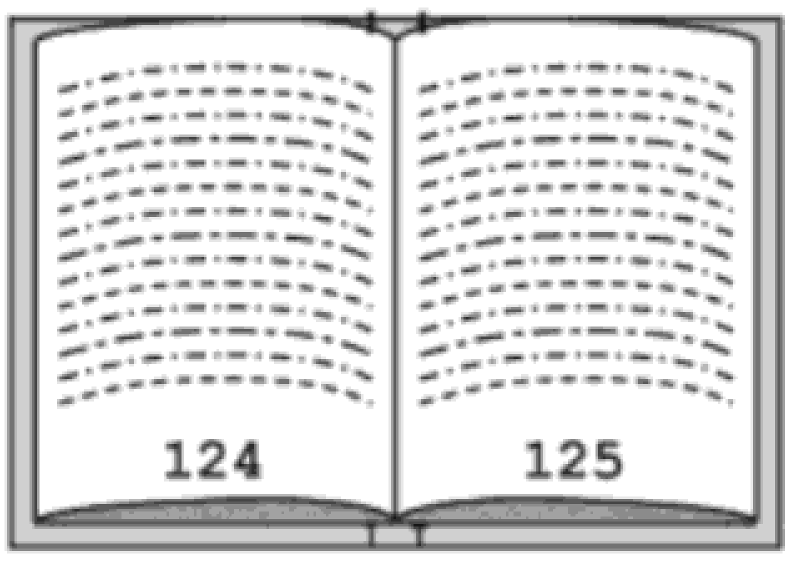


Pages 500 and 501 were both blank, so Frank knew the last written page was on or before 499—his new upper bound. After another midpoint computation, he flipped to page 250. Again it was blank.

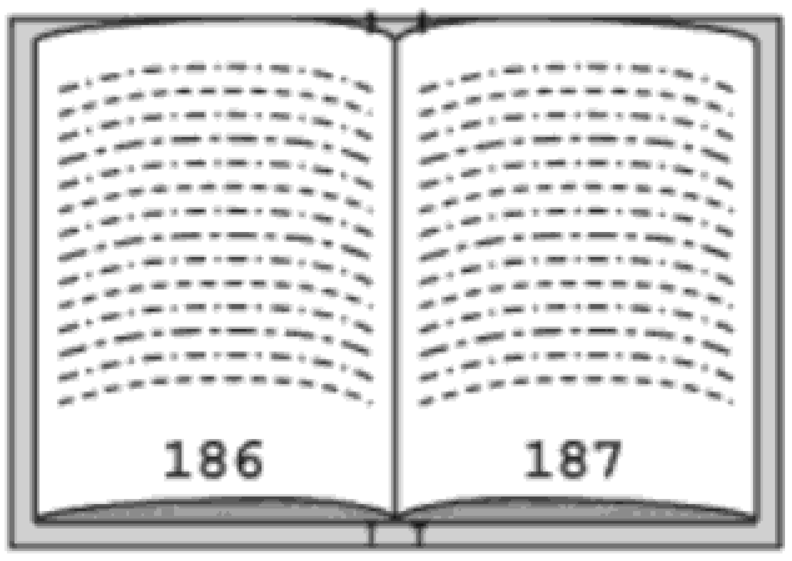


"Looks like a new book," added Notation. "Good thing you didn't keep going from the back."

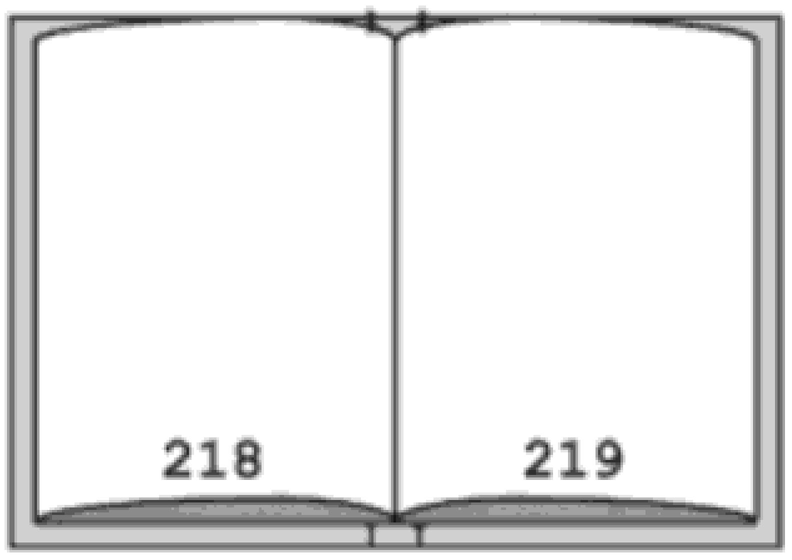
Frank didn't bother replying. With a lower bound of 1 and an upper bound of 249, he computed a midpoint of 125. This time he found writing, so he adjusted his lower bound accordingly to 125.



"187," supplied Notation before Frank could finish the midpoint computation in his head. He turned to 187, again finding writing and adjusting his lower bound.



"218," said Notation. The pages were blank, so Frank adjusted his lower and upper bounds to 187 and 217, respectively.



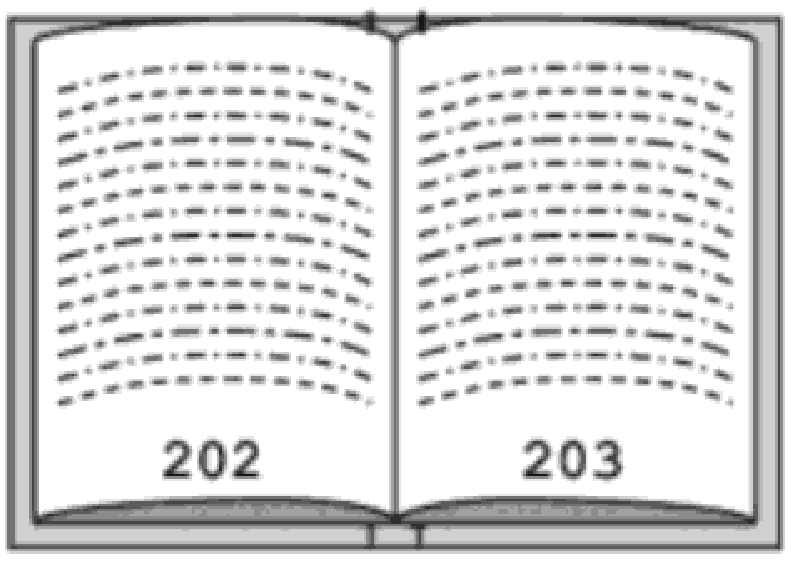
"202," said Notation before Frank had even finished adding the upper and lower bounds.

"How are you doing that so fast?" asked Frank.

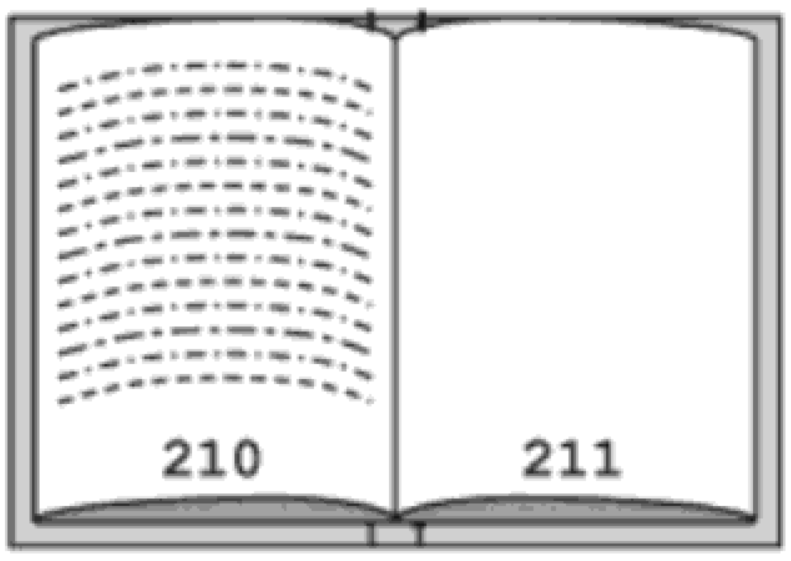
"Practice," she replied. "We used to have binary search competitions at the academy whenever we needed a break from studying. I was undefeated."

Frank shook his head. "Sounds like a wild time," he muttered.

Pages 202 and 203 were filled. "210," supplied Notation.



At page 210, they finally found the last entry, detailing the *Retry Loop*'s last voyage. "What now?" asked Notation.



"We search for an interesting package or port. On the last voyage, they made about 70 entries. We'll have to scan through them."

"Exhaustive search?" asked Notation. "Can't we use something more efficient? Aren't the entries sorted by pickup and delivery time?"

"The sorting doesn't help us here," answered Frank. "We don't know the time. Sorted data only helps when it's sorted by a useful dimension. They didn't bother to sort by suspiciousness. Go figure."

"Oh. It's the Weather Records problem," said Notation.

"What problem?" asked Frank.

"It's an illustration of how sorting data by the wrong value doesn't help a search," explained Notation. "Professor Drecker gave the example of finding the coldest day in the last 10 years. If the logs are sorted by day, you can use binary search to efficiently find any specific day. But that doesn't help us to find the coldest day, so we'd still have to scan through all the data. I hadn't expected to see such a clear example outside of class, though."

“Welcome to the real world,” said Frank. “Out here, you have to check when the structure in the data is helpful to you and when it isn’t. Don’t worry, it’s a common rookie mistake.”

He could see Notation bristle at his words and tried not to enjoy her reaction too much. Every rookie came out of the academy thinking they knew it all, and every one had a lot to learn. Notation was getting off easy with a lecture. His own education with binary search had involved hours of scooping through barrels of pig waste while he questioned his career choice.

After about three minutes, they found their only clue. The *Retry Loop* had recently made two suspicious stops, at the Port of Mudwall and Frayed Cable Island. Even for smugglers, these were strange destinations. The Port of Mudwall boasted little trade beyond its outlying mud farms. And Frayed Cable Island was even more desolate; the small, rocky island possessed only a single building—the now abandoned Iron Ring Prison.

“There,” said Frank, pointing. “That’s where they took your documents. Either the Port of Mudwall or Frayed Cable Island. They probably dropped the documents off at one and picked up the payment at the other.”

“How do you know?” asked Notation. She looked skeptical. “Shouldn’t we consider all the ports as—”

Frank cut her off. “No time to check them all.” He didn’t elaborate. He was using his own brand of algorithm now, the heuristic searches that had gotten him in trouble with the captain in the past. But he had a gut feeling, and Frank had learned to trust his gut.

“Are you sure that—” Notation began, but was cut off by noises above them.

Frank couldn’t make out the words, but he could recognize the tone clearly. Trouble was on its way.

POLICE ALGORITHMS 101: BINARY SEARCH II
Excerpts from Professor Drecker's Lecture

The key to efficient algorithms is information. In the case of binary search, we require that the data be sorted and that we have information on how that data is sorted. In order to rule out (or prune) large regions of the search space, the algorithm must be able to guarantee that the target value can't be in that region. We can only do that if we know how the values behave as we move along the array. In computational problems, we say the array is sorted if all its values are arranged in increasing (or decreasing) order.

However, just because the data is sorted by *one* dimension doesn't mean that you will be able to binary search along another dimension. Say you're searching an accounting ledger for clues. Ledgers are sorted by transaction number, which indicates when the transaction was *recorded*. This means that the transaction number for each entry will be less than the transaction number for the following entry. If the current entry has a transaction number of 105, we know that all entries before it will have transaction numbers less than 105 and all entries after it will have transaction numbers greater than 105.

101	August 16	Zed's Coffee	8.00
102	August 15	Bob's Pizza	20.00
103	August 15	Wands and More	150.00
104	August 15	Spell Shoppe	100.00
105	August 16	Zed's Coffee	8.00
106	August 16	Spell Shoppe	50.00
107	August 17	Zed's Coffee	8.00
108	August 17	Hospital	250.00

continued

However, this also means that the entries in other fields, such as the actual date of the transaction, the mechant's name, or the transaction amount, are *not* in sorted order. What if you're interested in finding transactions above a certain suspicious amount or with a known weapons merchant? Does the sorting help you here? No, you would still find yourself using an exhaustive linear search. Knowing that transaction 105 was at Zed's Coffee doesn't tell you anything about the merchants or amounts in transactions before or after that one.

Similarly, if you sorted the ledger in increasing order of transaction amount, this would allow you to quickly find all transactions costing $250, but it would not help you search for a given transaction date, ID, or merchant.

— 7 —
Adapting Algorithms for a Daring Escape

Heavy footsteps pounded the wooden deck above. Frank glanced around, assessing their limited options. The only hatch led up to the deck and the new arrivals. The hold itself was nearly empty, the crew having unloaded the cargo upon arriving in Usb. Trying to hide here would amount to standing in a corner and whispering "You can't see me."

As Frank listed and discarded every possible option, including the rarely effective ploy of lying down and playing dead, he saw Notation pull out her badge and stand at attention.

"What are you thinking?" he hissed.

"I am an officer of the law on an official investigation," explained Notation.

Frank shook his head in disbelief. "The 'Stop in the name of the law' routine isn't about to work here. Or most places, for that matter. We're on a smuggler's ship, investigating the theft of police property. No one on the force even knows you're here, do they? And I'd be willing to bet whoever is coming through that door knows that."

Notation opened her mouth to argue, but paused and closed it again. She slid her badge back into her jacket as a stream of large and surprisingly well-dressed thugs poured through the door. They