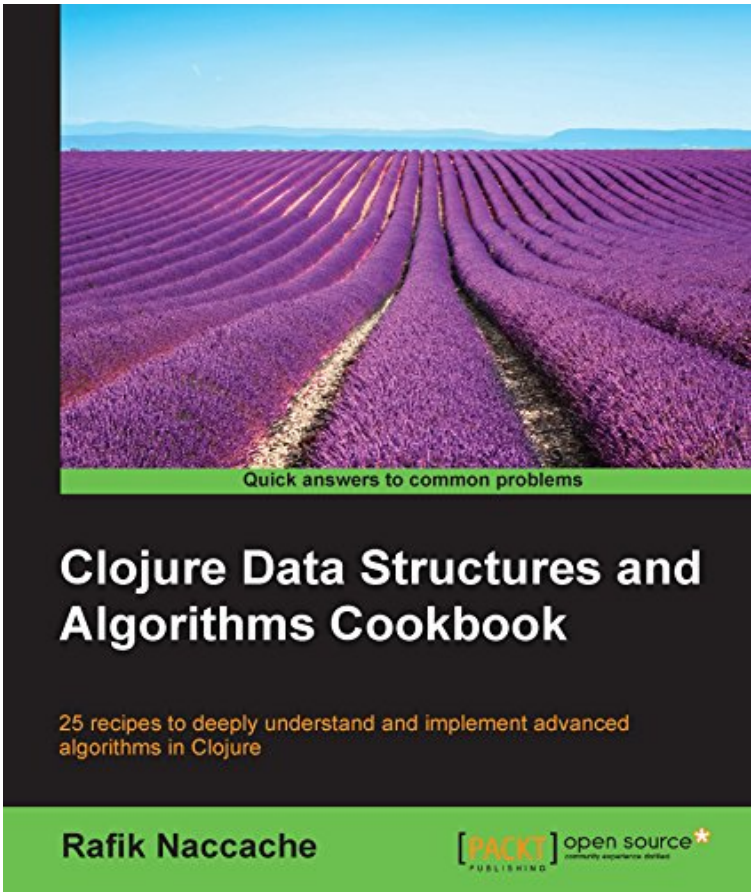
Clojure Data Structures and Algorithms Cookbook

Von Rafik Naccache
*audiobook | *ebooks | Download PDF | ePub | DOC*



Produktinformation -Verkaufsrang: #813185 in eBooksVerffentlicht am: 2015-09-03Erscheinungsdatum: 2015-09-03File Name: B00YSIM3PU | File size: 64.Mb

Von Rafik Naccache : Clojure Data Structures and Algorithms Cookbook before purchasing it in order to gage whether or not it would be worth my time, and all praised Clojure Data Structures and Algorithms Cookbook:

KundenrezensionenHilfreichste Kundenrezensionen0 von 0 Kunden fanden die folgende Rezension hilfreich. Anspruchsvoll und gleichermaen praxisnahVon Manfred BerndtgenDer Begriff 'Datenstrukturen' ist eine Art Reizwort in der Welt funktionaler Programmiersprachen, basieren doch Datenstrukturen und Algorithmen in der klassischen (imperativen) Art auf der Mutabilitt von und Zuweisungen an Variablen. Bekanntermaen ist dies bei funktionalen Programmiersprachen nicht mglich bzw. wird dort nicht unterstzt. Chris Okasakis Dissertation (https://www.cs.cmu.edu/~rwh/theses/okasaki.pdf) diskutiert dieses Thema ausfhrlich; seitdem wurden weitere rein funktionale Datenstrukturen entwickelt, alle davon auf sehr hohem Niveau und nichts fr Anfnger in diesem Themengebiet. Kurz gesagt erfordert die Behandlung von Datenstrukturen und Algorithmen bei funktionaler Programmierung andere Anstze und Methoden als bei imperativen Sprachen.Weder erfindet Rafik Naccache das Rad neu noch entwickelt er neue funktionale Datenstrukturen in seinem Buch 'Clojure Data Structures and Algorithms Cookbook'. Stattdessen nutzt er den funktionalen Ansatz von Clojure zur Lsung anspruchsvoller Probleme mit praktischen Anwendungen. Sein Buch behandelt eine gre Flle von Algorithmen, beginnend bei klassischen

Datenstrukturen wie Arrays (Datenkompression, Fraktale, Stacks), Listen (doppelt verlinkte XOR-Listen, Shift-Reduce Parser, Random Access-Listen), mehrere Varianten von Bumen, Entscheidungs- und Optimierungsprobleme (Recommendation Engine, optimaler Pfad, Zusammenfassen von Texten), Logische Programmierung, Kommunikationstools sowie inhrent funktionale Entwicklungen (rekursiver Abstieg, Firewallsimulator, Unifikation). Die ausgewhlten Themen sind alle aktuell und selbst wenn der Leser dieses Buch nicht von Anfang bis Ende durchliest, ist seine Lektre uerst lohnend.Die Struktur des Buches untersttzt das schnelle Durcharbeiten ohne dabei auf Details zu verzichten. Nach einer Einfhrung zu jedem Thema und seiner theoretischen Basis diskutiert der Autor seine Implementierung und gibt ein Anwendungsbeispiel des zuvor Entwickelten. Auf diese Weise wird der komplette Entwicklungsprozess von Theorie bis Testung abgedeckt. Weitere Informationen verweisen auf die (meist wissenschaftlichen) Arbeiten der jeweiligen Autoren der verwendeten Algorithmen. Naccaches Code funktioniert und ist erstaunlich knapp; ein Hinweis darauf, da die ausgewhlten Themen sehr gut durch funktionale Programmierung abgedeckt werden knnen.Ein Sonntagsspaziergang ist dieses Buch trotzdem nicht: der Leser sollte mindestens fortgeschrittene Kenntnisse in Clojure-Programmierung haben, ebenso ein starkes Interesse an den behandelten Themen und Algorithmen. Dieses Buch hat mich stark an die 'Numerical Recipes'-Reihe (https://en.wikipedia.org/wiki/Numerical_Recipes) erinnert, geschrieben fr Clojure-Programmiere in einer exakten und knappen Sprache. Empfehlenswert.

KurzbeschreibungData-structures and algorithms often cross your path when you compress files, compile programs, access databases, or simply use your favourite text editor. Understanding and implementing them can be daunting. Curious learners and industrial developers can find these complex, especially if they focus on the detailed implementation of these data structures.Clojure is a highly pragmatic and expressive language with efficient and easy data manipulation capabilities. As such, it is great for implementing these algorithms. By abstracting away a great share of the unnecessary complexity resulting from implementation, Clojure and its contrib libraries will help you address various algorithmic challenges, making your data exploration both profitable and enjoyable.Through 25 recipes, you'll explore advanced algorithms and data-structures, well served by a sound Clojure implementation.This book opens with an exploration of alternative uses of the array data-structure, covering LZ77 compression, drawing fractals using Pascal's triangles, simulating a multi-threaded program execution, and implementing a call-stack winding and un-winding operations.The book elaborates on linked lists, showing you how to construct doubly linked ones, speed up search times over the elements of such structures, use a linked-list as the foundation of a shift-reduce parser, and implement an immutable linked-list using skew binary numbers representation.After that, the tree data-structure is explored, focusing on building self-balancing Splay Trees, designing a B-Tree backing-up an efficient key-value data-store, constructing an undo capable Rope, and showing how Tries can make for an auto-completing facility.Next, some optimization and machine learning techniques are discussed, namely for building a co-occurrence-based recommendation engine, using branch-and-bound to optimize integral cost and profit problems, using Dijkstra's algorithm to determine optimal paths and summarizing texts using the LexRank algorithm.Particular attention is given to logic programming, you will learn to use this to discover interesting relations between social website data, by designing a simple type inferencer for a mini Java-like language, and by building a simple checkers game engine.Asynchronous programming will be addressed and you will design a concurrent web-crawler, an efficient Clojure source files indexer, and an online taxi booking platform.Finally, youll discover how higher order functions can be used in Clojure to achieve compoundable and reusable transformations including tweaking nrepl to deliver evaluations by e-mail and simulating a firewall or a mail inbox processor using new Clojure 1.7's transducers.KurzbeschreibungData-structures and algorithms often cross your path when you compress files, compile programs, access databases, or simply use your favourite text editor. Understanding and implementing them can be daunting. Curious learners and industrial developers can find these complex, especially if they focus on the detailed implementation of these data structures.Clojure is a highly pragmatic and expressive language with efficient and easy data manipulation capabilities. As such, it is great for implementing these algorithms. By abstracting away a great share of the unnecessary complexity resulting from implementation, Clojure and its contrib libraries will help you address various algorithmic challenges, making your data exploration both profitable and enjoyable.Through 25 recipes, you'll explore advanced algorithms and data-structures, well served by a sound Clojure implementation.This book opens with an exploration of alternative uses of the array data-structure, covering LZ77 compression, drawing fractals using Pascal's triangles, simulating a multi-threaded program execution, and implementing a call-stack winding and un-winding operations.The book elaborates on linked lists, showing you how to construct doubly linked ones, speed up search times over the elements of such structures, use a linked-list as the foundation of a shift-reduce parser, and implement an immutable linked-list using skew binary numbers representation.After that, the tree data-structure is explored, focusing on building self-balancing Splay Trees, designing a B-Tree backing-up an efficient key-value data-store, constructing an undo capable Rope, and showing how Tries can make for an auto-completing

facility.Next, some optimization and machine learning techniques are discussed, namely for building a co-occurrence-based recommendation engine, using branch-and-bound to optimize integral cost and profit problems, using Dijkstra's algorithm to determine optimal paths and summarizing texts using the LexRank algorithm.Particular attention is given to logic programming, you will learn to use this to discover interesting relations between social website data, by designing a simple type inferencer for a mini Java-like language, and by building a simple checkers game engine.Asynchronous programming will be addressed and you will design a concurrent web-crawler, an efficient Clojure source files indexer, and an online taxi booking platform.Finally, youll discover how higher order functions can be used in Clojure to achieve compoundable and reusable transformations including tweaking nrepl to deliver evaluations by e-mail and simulating a firewall or a mail inbox processor using new Clojure 1.7's transducers.ber den Autor und weitere MitwirkendeRafik Naccache Rafik Naccache is a Tunisian, who is experienced in software architecture and is an emergent technology enthusiast. He earned his bachelor's degree in computer science engineering from the University of Tunis in 2001. Rafik fell in love with Clojure back in 2012 and has been developing it professionally since 2013. He has occupied various positions in the telecom and banking sectors and has launched a few innovative start-ups on the Internet, in which he was able to deploy Clojure apps. He also founded the Tunisian Clojure enthusiasts community. He contributes to open source projects such as Cryogen (https://github.com/cryogen-project/cryogen/graphs/contributors) and Milestones (https://github.com/automagictools/milestones).