1. Record Nr. UNINA9910968733903321

Autore	Saha Suman
Titolo	Advanced data structures : theory and applications / / Suman Saha, Shailendra Shukla
Pubbl/distr/stampa	Boca Raton : , : Chapman & Hall/CRC, , 2019
ISBN	0-429-94985-5 0-429-94984-7 0-429-48875-0
Edizione	[1st ed.]
Descrizione fisica	1 online resource (261 pages)
Disciplina	005.73
Soggetti	Data structures (Computer science)
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Nota di bibliografia	Includes bibliographical references and indexes.
Nota di contenuto	Cover -- Half Title -- Title Page -- Copyright Page -- Dedication -- Contents -- Foreword -- Acknowledgments -- Preface -- Authors -- Part 1: Theoretical Advancements -- 1. Introduction -- 1.1 Data Structure -- 1.2 Design of Data Structure -- 1.3 Analysis of Data Structure -- 1.4 Amortized Complexity -- 1.5 Computational Models -- 1.5.1 RAM model -- 1.5.2 Word RAM model -- 1.5.3 Cell-probe model of computation -- 1.6 Bounds of Fundamental Data Structures -- 1.7 Lazy Delete -- 1.8 Organization of Part I -- 1.9 Exercises -- 2. O(1)Search by Hashing -- 2.1 Basic Hashing -- 2.1.1 Hash function -- 2.1.2 Load factor -- 2.1.3 Collision resolution -- 2.2 Perfect Hashing -- 2.2.1 Construction -- 2.2.2 Remarks -- 2.3 Universal Hashing -- 2.3.1 Important properties -- 2.3.2 Mathematical guarantees -- 2.4 Cuckoo Hashing -- 2.4.1 Operations -- 2.4.2 Bipartite graph of cuckoo hashing -- 2.5 Bloom Filters -- 2.5.1 Construction of bloom filter -- 2.5.2 Probability of false positives -- 2.5.3 Optimal values of parameters -- 2.6 Locality-Sensitive Hashing -- 2.6.1 Use in nearest neighbor search problem -- 2.7 Exercises -- 3. O(log(n)) Ordered Search (Trees and Lists) -- 3.1 Balanced Binary Search Trees (BSTs) -- 3.1.1 Height bound of balanced BST -- 3.2 Randomized BSTs -- 3.2.1 Static randomized BSTs -- 3.2.2 Dynamic randomized BSTs -- 3.2.3 Analysis of randomized BSTs -- 3.3 Splay Tree -- 3.3.1 Splaying -- 3.3.2 Splaying algorithms -- 3.3.3 Performance -- 3.4 Tango Tree --

3.4.1 Creation of tango tree -- 3.4.2 Tango analysis -- 3.5 Skiplists -- 3.5.1 Skipping -- 3.5.2 Dynamic updates -- 3.5.3 Probabilistic analysis of skiplist -- 3.6 Static and Dynamic Optimality -- 3.6.1 Search optimality in BST -- 3.6.2 Static optimality -- 3.6.3 Dynamic optimality -- 3.7 Exercises -- 4. Findset, Find Min, and Find Word -- 4.1 Disjoint Sets.
4.1.1 Operations on disjoint-set data structure -- 4.1.2 Representations of disjoint sets -- 4.1.3 Link-list representations of disjoint sets -- 4.1.4 Forest representations of disjoint sets -- 4.2 Binomial Heap -- 4.2.1 Creation and updates of binomial heap -- 4.2.2 Operations of Binomial Heap -- 4.2.3 Complexity -- 4.3 Fibonacci Heaps -- 4.3.1 Properties of a Fibonacci heap -- 4.3.2 Inserting, merging, cutting, and marking -- 4.3.3 Decreasing keys and delete-min operation -- 4.3.4 Algorithm for Fibonacci heaps -- 4.3.5 Amortized analysis for Fibonacci heaps -- 4.3.6 Tree size -- 4.4 Tries -- 4.4.1 Insertion -- 4.4.2 Searching -- 4.4.3 Deletion -- 4.4.4 Complexity -- 4.4.5 Compact trie -- 4.4.6 Patricia -- 4.4.7 Suffix tree -- 4.5 Inverted Index -- 4.5.1 Inverted index creation -- 4.5.2 Index compression -- 4.5.3 Key words search -- 4.6 Exercises -- Part 2: Evolving Paradigms -- 5. Evolving Paradigms of Data Structures -- 5.1 Geometric Queries -- 5.2 I/O Complexities -- 5.3 Communication Complexities -- 5.4 Large Data Problem -- 5.5 Exercise -- 6. Spatial Data Structures -- 6.1 Range Search Trees -- 6.1.1 Construction -- 6.1.2 Range query search -- 6.2 KD Trees -- 6.2.1 Creation of KD tree -- 6.2.2 Range search in KD tree -- 6.2.3 Nearest neighbor search in KD tree -- 6.3 Quadtree -- 6.3.1 Inserting data into a quadtree -- 6.3.2 Properties of quadtree -- 6.3.3 Region quadtree -- 6.3.4 Point quadtree -- 6.4 R Tree -- 6.4.1 Indexing structure of R tree -- 6.4.2 Search in R tree -- 6.4.3 Dynamic update of R tree -- 6.5 Exercises -- 7. Temporal Data Structures -- 7.1 Partial Persistence -- 7.1.1 Partial persistence -- 7.1.2 Full persistence -- 7.1.3 Confluent persistence -- 7.1.4 Functional persistence -- 7.2 Retroactivity -- 7.2.1 Decomposable search problem -- 7.3 Exercises -- 8. External Memory Data Structures -- 8.1 Input/Output (I/O) Model.
8.2 Cache Oblivious Algorithms -- 8.2.1 Cache aware model -- 8.2.2 Cache oblivious model -- 8.3 B, B+ Tree -- 8.3.1 Searching -- 8.3.2 Insertion -- 8.3.3 Removal -- 8.3.4 Amortized analysis of B trees -- 8.3.5 B+ tree -- 8.4 (a,b) Tree -- 8.4.1 Insertion -- 8.4.2 Deletion -- 8.5 Buffer Tree -- 8.6 Exercises -- 9. Distributed Data Structures (DDSs) -- 9.1 Descriptions of Structures -- 9.1.1 Properties of DDS -- 9.2 Distributed Hashing -- 9.2.1 Structure of distributed hashing -- 9.3 Distributed Trees -- 9.3.1 Construction of distributed BST -- 9.3.2 Insertion -- 9.3.3 Deletion -- 9.3.4 Rotation -- 9.4 Skip Graphs -- 9.4.1 Design -- 9.4.2 Search -- 9.4.3 Insertion -- 9.4.4 Deletion -- 9.4.5 Correctness and concurrency -- 9.5 Exercises -- 10. Synopsis Data Structures -- 10.1 Data Synopsis -- 10.1.1 Synopsis methods -- 10.1.2 Application -- 10.2 Sampling -- 10.2.1 Sampling technique -- 10.2.2 Reservoir sampling -- 10.2.3 Sampling with updates -- 10.2.4 Sliding window sampling -- 10.3 Sketching -- 10.3.1 Count-min sketches -- 10.4 Fingerprint -- 10.4.1 Fingerprinting scheme of Rabin -- 10.5 Wavelets -- 10.5.1 Wavelet decomposition -- 10.6 Exercises -- Part 3: Recent Applications -- 11. Introduction to Applications -- 11.1 Various Domain Applications -- 11.2 Project -- 12. Applications to Cryptography -- 12.1 MD5 -- 12.1.1 Password hashing -- 12.2 Secure Socket Layers (SSLs) -- 12.2.1 Data structure of open SSL -- 12.3 Block Chains -- 12.4 Digital Signature -- 12.5 Projects -- 13. Application to IR and WWW -- 13.1 Crawl Frontier -- 13.2 Posting List Intersection -- 13.3 Text Retrieval from Inverted Index -- 13.4 Auto Complete Using

Tries -- 13.5 Projects -- 14. Applications to Data Science -- 14.1 Heavy Hitters and Count-Min Structures -- 14.2 Approximate Nearest Neighbor Searches -- 14.2.1 Approximate nearest neighbor. 14.2.2 Locality-sensitive hashing (LSH) -- 14.3 Low Rank Approximation by Sampling -- 14.3.1 Nystrom approximation -- 14.3.2 Random sketching -- 14.4 Near-Duplicate Detection by Min Hashing -- 14.5 Projects -- 15. Application to Network and IOT -- 15.1 Click-Stream Processing Using Bloom Filters -- 15.1.1 GBF Algorithm -- 15.2 Fast IP-Address Lookup Using Tries -- 15.3 Integrity Verification: Cloud and IOT Data -- 15.4 Projects -- 16. Applications to Systems -- 16.1 Queue Spilling -- 16.2 Completely Fair Schedulers in Kernels -- 16.2.1 CFS internals -- 16.3 Distributed Caching -- 16.4 Data Structures for Building File Systems -- 16.5 Projects -- 17. Applications to Databases -- 17.1 Database Problems -- 17.1.1 Searching sorted files -- 17.1.2 Index for first search -- 17.1.3 Insertion deletion in database -- 17.2 B and B+ Trees for Database Creation and Block Search -- 17.2.1 Applications of B trees in databases and file systems -- 17.3 CouchDB -- 17.4 Bloomjoins -- 17.5 Projects -- 18. Applications to Images and Graphics -- 18.1 R Trees for Map Searches -- 18.1.1 R trees for mapping -- 18.1.2 Insertion -- 18.1.3 Deletion -- 18.1.4 Search -- 18.2 Spatial Proximity in GIS -- 18.2.1 GIS objects -- 18.2.2 Data access in GIS -- 18.2.3 Computational requirements -- 18.2.4 Solution using k-d tree -- 18.3 Ray Shooting -- 18.3.1 Rays -- 18.3.2 Camera-ray intersections -- 18.3.3 Shadow rays -- 18.3.4 Reflection rays -- 18.3.5 Transmission rays -- 18.3.6 Recursive ray tracing -- 18.3.7 Ray intersection -- 18.3.8 Bounding volume hierarchies -- 18.4 Data Structures Used in Ray Shooting -- 18.4.1 Octrees -- 18.4.2 KD trees -- 18.4.3 BSP trees -- 18.4.4 Uniform grids -- 18.4.5 Hierarchical grids -- 18.5 Projects -- Bibliography -- Index.

Sommario/riassunto

Advanced data structures is a core course in Computer Science which most graduate program in Computer Science, Computer Science and Engineering, and other allied engineering disciplines, offer during the first year or first semester of the curriculum. The objective of this course is to enable students to have the much-needed foundation for advanced technical skill, leading to better problem-solving in their respective disciplines. Although the course is running in almost all the technical universities for decades, major changes in the syllabus have been observed due to the recent paradigm shift of computation which is more focused on huge data and internet-based technologies. Majority of the institute has been redefined their course content of advanced data structure to fit the current need and course material heavily relies on research papers because of nonavailability of the redefined text book advanced data structure. To the best of our knowledge well-known textbook on advanced data structure provides only partial coverage of the syllabus. The book offers comprehensive coverage of the most essential topics, including: Part I details advancements on basic data structures, viz., cuckoo hashing, skip list, tango tree and Fibonacci heaps and index files. Part II details data structures of different evolving data domains like special data structures, temporal data structures, external memory data structures, distributed and streaming data structures. Part III elucidates the applications of these data structures on different areas of computer science viz, network, www, DBMS, cryptography, graphics to name a few. The concepts and techniques behind each data structure and their applications have been explained. Every chapter includes a variety of Illustrative Problems pertaining to the data structure(s) detailed, a summary of the technical content of the chapter and a list of Review

Questions, to reinforce the comprehension of the concepts. The book could be used both as an introductory or an advanced-level textbook for the advanced undergraduate, graduate and research programmes which offer advanced data structures as a core or an elective course. While the book is primarily meant to serve as a course material for use in the classroom, it could be used as a starting point for the beginner researcher of a specific domain.