1.	Record Nr.	UNINA9910900171003321
	Autore	Mailund Thomas
	Titolo	The Joys of Hashing : Hash Table Programming with C / / by Thomas Mailund
	Pubbl/distr/stampa	Berkeley, CA : , : Apress : , : Imprint : Apress, , 2024
	ISBN	979-88-6880-826-5
	Edizione	[2nd ed. 2024.]
	Descrizione fisica	1 online resource (225 pages)
	Disciplina	005.45
	Soggetti	Compilers (Computer programs) Computer programming Compilers and Interpreters Programming Techniques
	Lingua di pubblicazione	Inglese
	Formato	Materiale a stampa
	Livello bibliografico	Monografia
	Note generali	Includes index.
	Nota di contenuto	1. The Joys of Hashing -- 2. Hash Keys, Indices and Collisions -- 3. Collision Resolution, Load Factor and Performance -- 4. Resizing -- 5. Adding Application Keys and Values -- 6. Heuristic Hash Functions -- 7. Universal Hashing -- 8. Conclusions.
	Sommario/riassunto	Build working implementations of hash tables, written in the C programming language. The new and revised Second Edition has been enhanced with simplified code and more comprehensive explanations, aimed at improving clarity and comprehension while exploiting the latest features in C programming. This book starts with simple first attempts devoid of collision resolution strategies. It then progresses through improvements and extensions illustrating different design ideas and approaches, followed by experiments to validate the choices. You will learn how hash tables, when implemented and used appropriately, are exceptionally efficient data structures for representing sets and lookup tables, providing low overhead, constant time, insertion, deletion, and lookup operations. The Joys of Hashing walks you through the implementation of efficient hash tables and the pros and cons of different design choices when building tables. The source code used in the book is available on GitHub for your re-use and experiments. What You Will Learn Master the basic ideas behind

hash tables Carry out collision resolution, including strategies for handling collisions and their consequences for performance Resize or grow and shrink tables as needed Store values by handling when values must be stored with keys to make general sets and maps.