array reduction 3 hackerrank solution

array reduction 3 hackerrank solution is a common search query among programmers and coding enthusiasts looking to solve one of HackerRank's challenging problems. In this comprehensive article, we will provide an in-depth exploration of the Array Reduction 3 problem, walking you through the problem statement, constraints, and optimal approaches to formulating a solution. Readers will gain a clear understanding of the problem's requirements, discover effective algorithms for array reduction, and see step-by-step explanations, including code snippets. We will also discuss common pitfalls, optimization tips, and frequently asked questions related to this topic. Whether you are preparing for technical interviews or aiming to boost your problem-solving skills, this article will equip you with everything you need to confidently tackle the array reduction 3 hackerrank solution.

Understanding the Array Reduction 3 Problem

The Array Reduction 3 problem on HackerRank challenges programmers to reduce a given array to a specific form based on defined operations. Typically, these problems require performing certain steps repeatedly to transform the array until it meets a particular condition. The main objective is to either minimize the number of operations or achieve the desired array structure efficiently. Understanding the precise requirements is crucial before attempting any solution, as this lays the foundation for developing an effective algorithm.

Problem Statement Overview

In the array reduction 3 hackerrank solution, the problem often involves being given an array of integers and a set of allowed operations. The goal is usually to reduce the array to a single value or a simplified form, such as making all elements equal, by applying the permitted operations. The exact operations might include removing elements, performing mathematical transformations, or merging

values according to specific rules.

Real-World Applications

Array reduction techniques are not only academic exercises but also have practical applications in data processing, optimization algorithms, and system design. For example, reducing redundancy in data arrays or optimizing resource allocation in distributed computing can benefit from similar logic used in solving such problems.

Key Constraints and Requirements

Understanding the constraints is essential for crafting an optimal array reduction 3 hackerrank solution. Constraints often define the size of the array, the range of element values, and the permissible operations. These boundaries affect the choice of algorithm and its efficiency.

Typical Constraints

- Array length: Usually up to 10^5 or higher
- Element range: Can include negative or positive integers, sometimes bounded by $\pm 10^9$
- Operation limits: Restrictions on the number or type of operations allowed
- Time complexity: Solutions often need to run in O(n log n) or better

Input and Output Format

The problem specifies the format for receiving input and producing output. Typically, the input consists of the array’s length followed by its elements, and the output is the minimum number of operations, the final reduced value, or a transformed array, depending on the problem’s objective.

Popular Approaches for Array Reduction

Several strategies exist for tackling array reduction problems, each with its own advantages. Selection of the optimal approach depends on the problem’s unique constraints and requirements. Here are some common methods used by experienced programmers.

Greedy Algorithms

A greedy approach involves making the optimal choice at each step with the hope of finding the global optimum. For array reduction, this might mean always selecting the largest or smallest element for reduction, depending on the allowed operations.

Sorting and Frequency Analysis

Sorting the array or analyzing the frequency of elements can reveal patterns that simplify reduction. For instance, making all elements equal to the most frequent value often leads to the minimum number of operations.

Dynamic Programming

For more complex constraints, dynamic programming can be used to break the problem into smaller subproblems. This technique is particularly useful when the operations have dependencies on previous steps.

Step-by-Step Solution Explanation

To provide a clear array reduction 3 hackerrank solution, it’s important to break down the process into manageable steps. Below is a generic approach that can be adapted based on the specific problem variation.

1. Read the input array and identify the objective (e.g., make all elements equal).
2. Analyze the frequency of each element using a hash map or array.
3. Determine the target value for reduction, often the most frequent element.
4. Calculate the minimum number of operations required by comparing each element to the target.
5. Implement the reduction using allowed operations, updating the array as needed.
6. Output the result as per the problem’s requirements.

Example Scenario

Consider an array [1, 2, 2, 3, 3, 3]. The most frequent element is 3. Reducing the array to all 3’s would require changing the other elements (1 and 2) to 3, resulting in three operations. This straightforward approach leverages frequency analysis for an efficient solution.

Sample Code and Walkthrough

Implementing the array reduction 3 hackerrank solution usually involves concise and efficient code. Below is a Python example that demonstrates the frequency analysis approach.

Python Solution Example:

```
def array_reduction(arr):
freq = {}
for num in arr:
freq[num] = freq.get(num, 0) + 1
max_freq = max(freq.values())
min_operations = len(arr) - max_freq
return min_operations

# Sample usage
arr = [1, 2, 2, 3, 3, 3]
print(array_reduction(arr))  # Output: 3
```

This code calculates the minimum number of operations required to reduce the array to all equal elements by changing non-target elements.

Optimization Techniques and Best Practices

Optimizing your array reduction 3 hackerrank solution can lead to faster execution and better performance, especially for large arrays. Here are some best practices:

- Use efficient data structures such as hash maps for counting frequencies.
- Minimize unnecessary iterations by breaking early when possible.
- Leverage built-in functions for sorting and counting to reduce code complexity.
- Test edge cases, such as empty arrays or arrays with all identical elements.

- Profile your solution to identify bottlenecks and improve accordingly.

Common Mistakes and How to Avoid Them

Many programmers encounter pitfalls when attempting array reduction problems. Being aware of common mistakes can save time and frustration.

- Failing to consider all possible allowed operations.
- Overcomplicating the solution when a simpler approach suffices (such as frequency analysis instead of dynamic programming).
- Ignoring edge cases, such as arrays with a single element or all elements already equal.
- Not optimizing for time and space complexity, leading to timeouts on large inputs.

Tips for Success

Carefully read the problem statement, validate your logic on multiple test cases, and always analyze the constraints before coding. Practicing similar problems can also help reinforce these skills.

Conclusion and Additional Tips

Solving the array reduction 3 hackerrank solution requires a solid grasp of array manipulation techniques, algorithmic efficiency, and attention to detail. By understanding the problem requirements, applying optimal strategies, and avoiding common mistakes, programmers can develop robust solutions. Continuing to practice and refine these skills will lead to greater success in coding challenges and technical interviews.

Q: What is the primary objective of the array reduction 3 hackerrank solution?

A: The main goal is to reduce a given array to a specific form, such as making all elements equal or achieving a particular arrangement, using a set of allowed operations with the minimum number of steps.

Q: Which algorithm is most commonly used for the array reduction 3 hackerrank solution?

A: Frequency analysis combined with greedy algorithms is often the most effective approach, as it minimizes the number of operations by targeting the most frequent element.

Q: Why is analyzing the frequency of elements important in array reduction?

A: Frequency analysis helps identify the optimal target value for reduction, allowing you to minimize the number of operations required to achieve the desired array state.

Q: What data structures are recommended for implementing this solution efficiently?

A: Hash maps (dictionaries) are recommended for counting element frequencies, as they provide fast lookup and update operations.

Q: Can sorting the array improve the performance of the solution?

A: Sorting can help in certain scenarios, but for most frequency-based reductions, directly counting frequencies is more efficient and avoids the O(n log n) overhead of sorting.

Q: What are some common pitfalls to avoid in array reduction problems?

A: Common mistakes include not considering all allowed operations, ignoring edge cases, and failing to optimize for large input sizes.

Q: How can you handle very large arrays efficiently in this problem?

A: Use space-efficient data structures, avoid unnecessary computations, and leverage optimized built-in functions to ensure your solution scales well.

Q: Is dynamic programming necessary for the array reduction 3 hackerrank solution?

A: In most cases, dynamic programming is not necessary unless the problem introduces additional complexities or dependencies between operations.

Q: What should you do if your solution times out on large test cases?

A: Re-examine your algorithm for time complexity improvements, use efficient data structures, and avoid redundant computations to optimize performance.

Q: Are there real-world applications for array reduction techniques?

A: Yes, array reduction strategies are used in data deduplication, optimization problems, and various systems where minimizing operations or resources is essential.

Array Reduction 3 Hackerrank Solution

Array Reduction 3 HackerRank Solution: A Comprehensive Guide

Are you grappling with the HackerRank challenge "Array Reduction 3"? This seemingly simple problem can be surprisingly tricky, leading many programmers down a path of inefficient solutions. This comprehensive guide provides a clear, step-by-step explanation of an efficient solution for Array Reduction 3, along with code examples in Python and the underlying logic to help you conquer this HackerRank challenge and improve your dynamic programming skills. We'll explore various approaches, analyze their time complexity, and provide insights into optimizing your code for maximum performance. Let's dive in!

Understanding the Problem: Array Reduction 3

The Array Reduction 3 problem on HackerRank presents you with an array of integers and asks you to find the minimum number of operations needed to reduce the array to a single element by repeatedly applying the operation of replacing any two adjacent elements with their absolute difference. This might sound straightforward, but finding the minimum number of operations requires a strategic approach. A brute-force method would quickly become computationally expensive for larger arrays.

Naive Approach and its Limitations

A naive approach might involve trying all possible combinations of adjacent element reductions. However, the number of possible combinations grows exponentially with the size of the array, making this approach impractical for larger inputs. The time complexity of such an approach would be O(2^n), where n is the array's length, rendering it highly inefficient.

Dynamic Programming: The Key to Efficiency

The most effective solution leverages dynamic programming. Dynamic programming solves complex problems by breaking them down into smaller, overlapping subproblems, solving each subproblem only once, and storing their solutions to avoid redundant computations. This significantly improves efficiency compared to brute-force approaches.

Implementing the Dynamic Programming Solution (Python)

Here's a Python implementation of a dynamic programming solution for Array Reduction 3:

```python
def array_reduction_3(arr):
n = len(arr)
if n == 1:
return 0

dp = {} # Dictionary for memoization

def solve(i, j):
if i == j:
return 0
if (i, j) in dp:
return dp[(i, j)]

min_ops = float('inf')
for k in range(i, j):
ops = 1 + solve(i, k) + solve(k + 1, j) + abs(arr[i] - arr[j])
min_ops = min(min_ops, ops)

dp[(i, j)] = min_ops
return min_ops

return solve(0, n - 1)
```

```
# Example usage
arr = [1, 2, 3, 4, 5]
result = array_reduction_3(arr)
print(f"Minimum operations: {result}")
```

Understanding the Code:

1. `dp = {}`: This dictionary stores the results of subproblems, avoiding redundant calculations (memoization).
2. `solve(i, j)`: This recursive function calculates the minimum operations needed to reduce the subarray `arr[i:j+1]`.
3. Base Case: If `i == j` (subarray with one element), it returns 0 operations.
4. Memoization: Checks if the result for `(i, j)` is already in `dp`. If so, it returns the stored value.
5. Iteration: Iterates through possible splitting points `k` to find the minimum operations.
6. Recursive Calls: Recursively calls `solve` for the left and right subarrays.
7. Minimum Operations: Updates `min_ops` with the minimum operations found.
8. Storing Result: Stores the minimum operations for `(i, j)` in `dp`.
9. Return Value: Returns the minimum operations for the entire array.

Time and Space Complexity Analysis

The dynamic programming solution significantly improves the time complexity. While the recursive nature might seem concerning, the memoization technique ensures that each subproblem is solved only once. The time complexity becomes O(n^3), where n is the length of the array, a considerable improvement over the exponential complexity of the naive approach. The space complexity is O(n^2) due to the memoization dictionary.

Optimizations and Further Considerations

While the above dynamic programming solution is efficient, further optimizations might be possible depending on the specific constraints of the HackerRank problem. For example, exploring iterative dynamic programming instead of recursion could potentially reduce the overhead associated with function calls.

Conclusion

Solving the Array Reduction 3 problem efficiently requires a solid understanding of dynamic programming. By breaking down the problem into smaller, overlapping subproblems and using memoization to store the results, we can achieve a significantly improved time complexity compared

to naive approaches. The provided Python code offers a robust and efficient solution, allowing you to tackle this HackerRank challenge with confidence. Remember to always analyze the time and space complexity of your solutions to ensure optimal performance.

FAQs

1. What if the array contains only one element? If the array has only one element, the minimum number of operations is 0, as no reduction is needed.

2. Can this solution handle negative numbers in the array? Yes, the use of `abs()` in the calculation ensures that the solution correctly handles negative numbers.

3. What is the difference between memoization and tabulation in dynamic programming? Memoization is a top-down approach where we recursively solve subproblems and store the results, while tabulation is a bottom-up approach where we build a table of solutions iteratively.

4. Could a greedy approach solve this problem effectively? No, a greedy approach would not guarantee finding the minimum number of operations. A greedy strategy might make locally optimal choices that lead to a suboptimal global solution.

5. How can I further improve the space complexity of this solution? One potential optimization would be to use a more space-efficient data structure for memoization or to explore iterative dynamic programming techniques that might reduce the space requirements.

array reduction 3 hackerrank solution: *Guide to Competitive Programming* Antti Laaksonen, 2018-01-02 This invaluable textbook presents a comprehensive introduction to modern competitive programming. The text highlights how competitive programming has proven to be an excellent way to learn algorithms, by encouraging the design of algorithms that actually work, stimulating the improvement of programming and debugging skills, and reinforcing the type of thinking required to solve problems in a competitive setting. The book contains many “folklore” algorithm design tricks that are known by experienced competitive programmers, yet which have previously only been formally discussed in online forums and blog posts. Topics and features: reviews the features of the C++ programming language, and describes how to create efficient algorithms that can quickly process large data sets; discusses sorting algorithms and binary search, and examines a selection of data structures of the C++ standard library; introduces the algorithm design technique of dynamic programming, and investigates elementary graph algorithms; covers such advanced algorithm design topics as bit-parallelism and amortized analysis, and presents a focus on efficiently processing array range queries; surveys specialized algorithms for trees, and discusses the mathematical topics that are relevant in competitive programming; examines advanced graph techniques, geometric algorithms, and string techniques; describes a selection of more advanced topics, including square root algorithms and dynamic programming optimization. This easy-to-follow guide is an ideal reference for all students wishing to learn algorithms, and practice for programming contests. Knowledge of the basics of programming is assumed, but previous background in algorithm design or programming contests is not necessary. Due to the broad range of topics covered at various levels of difficulty, this book is suitable for both beginners and more experienced readers.

array reduction 3 hackerrank solution: Cracking the Coding Interview Gayle Laakmann McDowell, 2011 Now in the 5th edition, Cracking the Coding Interview gives you the interview preparation you need to get the top software developer jobs. This book provides: 150 Programming Interview Questions and Solutions: From binary trees to binary search, this list of 150 questions includes the most common and most useful questions in data structures, algorithms, and knowledge based questions. 5 Algorithm Approaches: Stop being blind-sided by tough algorithm questions, and learn these five approaches to tackle the trickiest problems. Behind the Scenes of the interview processes at Google, Amazon, Microsoft, Facebook, Yahoo, and Apple: Learn what really goes on during your interview day and how decisions get made. Ten Mistakes Candidates Make -- And How to Avoid Them: Don't lose your dream job by making these common mistakes. Learn what many candidates do wrong, and how to avoid these issues. Steps to Prepare for Behavioral and Technical Questions: Stop meandering through an endless set of questions, while missing some of the most important preparation techniques. Follow these steps to more thoroughly prepare in less time.

array reduction 3 hackerrank solution: *Introduction To Algorithms* Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, Clifford Stein, 2001 An extensively revised edition of a mathematically rigorous yet accessible introduction to algorithms.

array reduction 3 hackerrank solution: Problem Solving in Data Structures and Algorithms Using Java Hemant Jain, 2016-10-21 This book is about the usage of Data Structures and Algorithms in computer programming. Designing an efficient algorithm to solve a computer science problem is a skill of Computer programmer. This is the skill which tech companies like Google, Amazon, Microsoft, Adobe and many others are looking for in an interview. This book assumes that you are a JAVA language developer. You are not an expert in JAVA language, but you are well familiar with concepts of references, functions, lists and recursion. In the start of this book, we will be revising the JAVA language fundamentals. We will be looking into some of the problems in arrays and recursion too. Then in the coming chapter, we will be looking into complexity analysis. Then will look into the various data structures and their algorithms. We will be looking into a Linked List, Stack, Queue, Trees, Heap, Hash Table and Graphs. We will be looking into Sorting & Searching techniques. Then we will be looking into algorithm analysis, we will be looking into Brute Force algorithms, Greedy algorithms, Divide & Conquer algorithms, Dynamic Programming, Reduction, and Backtracking. In the end, we will be looking into System Design, which will give a systematic approach for solving the design problems in an Interview.

array reduction 3 hackerrank solution: Iterative Dynamic Programming Rein Luus, 2019-09-17 Dynamic programming is a powerful method for solving optimization problems, but has a number of drawbacks that limit its use to solving problems of very low dimension. To overcome these limitations, author Rein Luus suggested using it in an iterative fashion. Although this method required vast computer resources, modifications to his original schem

array reduction 3 hackerrank solution: *Optimized C++* Kurt Guntheroth, 2016-04-27 In today’s fast and competitive world, a program’s performance is just as important to customers as the features it provides. This practical guide teaches developers performance-tuning principles that enable optimization in C++. You’ll learn how to make code that already embodies best practices of C++ design run faster and consume fewer resources on any computer—whether it’s a watch, phone, workstation, supercomputer, or globe-spanning network of servers. Author Kurt Guntheroth provides several running examples that demonstrate how to apply these principles incrementally to improve existing code so it meets customer requirements for responsiveness and throughput. The advice in this book will prove itself the first time you hear a colleague exclaim, “Wow, that was fast. Who fixed something?” Locate performance hot spots using the profiler and software timers Learn to perform repeatable experiments to measure performance of code changes Optimize use of dynamically allocated variables Improve performance of hot loops and functions Speed up string handling functions Recognize efficient algorithms and optimization patterns Learn the strengths—and weaknesses—of C++ container classes View searching and sorting through an optimizer’s eye Make efficient use of C++ streaming I/O functions Use C++ thread-based

concurrency features effectively

array reduction 3 hackerrank solution: Elements of Programming Interviews Adnan Aziz, Tsung-Hsien Lee, Amit Prakash, 2012 The core of EPI is a collection of over 300 problems with detailed solutions, including 100 figures, 250 tested programs, and 150 variants. The problems are representative of questions asked at the leading software companies. The book begins with a summary of the nontechnical aspects of interviewing, such as common mistakes, strategies for a great interview, perspectives from the other side of the table, tips on negotiating the best offer, and a guide to the best ways to use EPI. The technical core of EPI is a sequence of chapters on basic and advanced data structures, searching, sorting, broad algorithmic principles, concurrency, and system design. Each chapter consists of a brief review, followed by a broad and thought-provoking series of problems. We include a summary of data structure, algorithm, and problem solving patterns.

array reduction 3 hackerrank solution: *Algorithms Unlocked* Thomas H. Cormen, 2013-03-01 For anyone who has ever wondered how computers solve problems, an engagingly written guide for nonexperts to the basics of computer algorithms. Have you ever wondered how your GPS can find the fastest way to your destination, selecting one route from seemingly countless possibilities in mere seconds? How your credit card account number is protected when you make a purchase over the Internet? The answer is algorithms. And how do these mathematical formulations translate themselves into your GPS, your laptop, or your smart phone? This book offers an engagingly written guide to the basics of computer algorithms. In Algorithms Unlocked, Thomas Cormen—coauthor of the leading college textbook on the subject—provides a general explanation, with limited mathematics, of how algorithms enable computers to solve problems. Readers will learn what computer algorithms are, how to describe them, and how to evaluate them. They will discover simple ways to search for information in a computer; methods for rearranging information in a computer into a prescribed order ("sorting"); how to solve basic problems that can be modeled in a computer with a mathematical structure called a "graph" (useful for modeling road networks, dependencies among tasks, and financial relationships); how to solve problems that ask questions about strings of characters such as DNA structures; the basic principles behind cryptography; fundamentals of data compression; and even that there are some problems that no one has figured out how to solve on a computer in a reasonable amount of time.

array reduction 3 hackerrank solution: Introduction to Algorithms Udi Manber, 1989 This book emphasizes the creative aspects of algorithm design by examining steps used in the process of algorithm development. The heart of the creative process lies in an analogy between proving mathematical theorems by induction and designing combinatorial algorithms. The book contains hundreds of problems and examples. It is designed to enhance the reader's problem-solving abilities and understanding of the principles behind algorithm design. 0201120372B04062001

array reduction 3 hackerrank solution: Competitive Programming 2 Steven Halim, Felix Halim, 2011

array reduction 3 hackerrank solution: Python Programming and Numerical Methods Qingkai Kong, Timmy Siauw, Alexandre Bayen, 2020-11-27 Python Programming and Numerical Methods: A Guide for Engineers and Scientists introduces programming tools and numerical methods to engineering and science students, with the goal of helping the students to develop good computational problem-solving techniques through the use of numerical methods and the Python programming language. Part One introduces fundamental programming concepts, using simple examples to put new concepts quickly into practice. Part Two covers the fundamentals of algorithms and numerical analysis at a level that allows students to quickly apply results in practical settings. - Includes tips, warnings and try this features within each chapter to help the reader develop good programming practice - Summaries at the end of each chapter allow for quick access to important information - Includes code in Jupyter notebook format that can be directly run online

array reduction 3 hackerrank solution: HTML5 Hacks Jesse Cravens, Jeff Burtoft, 2012-11-15 With 90 detailed hacks, expert web developers Jesse Cravens and Jeff Burtoft demonstrate intriguing uses of HTML5-related technologies. Each recipe provides a clear

explanation, screenshots, and complete code examples for specifications that include Canvas, SVG, CSS3, multimedia, data storage, web workers, WebSockets, and geolocation. You'll also find hacks for HTML5 markup elements and attributes that will give you a solid foundation for creative recipes that follow. The last chapter walks you through everything you need to know to get your HTML5 app off the ground, from Node.js to deploying your server to the cloud. Here are just a few of the hacks you'll find in this book: Make iOS-style card flips with CSS transforms and transitions Replace the background of your video with the Canvas tag Use Canvas to create high-res Retina Display-ready media Make elements on your page user-customizable with editable content Cache media resources locally with the filesystem API Reverse-geocode the location of your web app user Process image data with pixel manipulation in a dedicated web worker Push notifications to the browser with Server-Sent Events

array reduction 3 hackerrank solution: Ludic, Co-design and Tools Supporting Smart Learning Ecosystems and Smart Education Óscar Mealha, Matthias Rehm, Traian Rebedea, 2020-09-09 This book presents papers from the 5th International Conference on Smart Learning Ecosystems and Regional Development, which promotes discussions on R&D work, policies, case studies, entrepreneur experiences, with a particular focus on understanding the relevance of smart learning ecosystems for regional development and social innovation, and how the effectiveness of the relation of citizens and smart ecosystems can be boosted. The book explores how technology-mediated instruments can foster citizens' engagement with learning ecosystems and territories, providing insights into innovative human-centric design and development models/techniques, education/training practices, informal social learning, innovative citizen-driven policies, and technology-mediated experiences and their impact. As such, it will inspire the social innovation sectors and ICT, as well as economic development and deployment strategies and new policies for smarter proactive citizens.

array reduction 3 hackerrank solution: Java By Comparison Simon Harrer, Jörg Lenhard, Linus Dietz, 2018-03-22 Write code that's clean, concise, and to the point: code that others will read with pleasure and reuse. Comparing your code to that of expert programmers is a great way to improve your coding skills. Get hands-on advice to level up your coding style through small and understandable examples that compare flawed code to an improved solution. Discover handy tips and tricks, as well as common bugs an experienced Java programmer needs to know. Make your way from a Java novice to a master craftsman. This book is a useful companion for anyone learning to write clean Java code. The authors introduce you to the fundamentals of becoming a software craftsman, by comparing pieces of problematic code with an improved version, to help you to develop a sense for clean code. This unique before-and-after approach teaches you to create clean Java code. Learn to keep your booleans in check, dodge formatting bugs, get rid of magic numbers, and use the right style of iteration. Write informative comments when needed, but avoid them when they are not. Improve the understandability of your code for others by following conventions and naming your objects accurately. Make your programs more robust with intelligent exception handling and learn to assert that everything works as expected using JUnit5 as your testing framework. Impress your peers with an elegant functional programming style and clear-cut object-oriented class design. Writing excellent code isn't just about implementing the functionality. It's about the small important details that make your code more readable, maintainable, flexible, robust, and faster. Java by Comparison teaches you to spot these details and trains you to become a better programmer. What You Need: You need a Java 8 compiler, a text editor, and a fresh mind.That's it.

array reduction 3 hackerrank solution: The Handbook of Electronic Trading Joseph Rosen, 2009-06-18 This book provides a comprehensive look at the challenges of keeping up with liquidity needs and technology advancements. It is also a sourcebook for understandable, practical solutions on trading and technology.

array reduction 3 hackerrank solution: Programming Challenges Steven S Skiena, Miguel A. Revilla, 2006-04-18 There are many distinct pleasures associated with computer programming.

Craftsmanship has its quiet rewards, the satisfaction that comes from building a useful object and making it work. Excitement arrives with the flash of insight that cracks a previously intractable problem. The spiritual quest for elegance can turn the hacker into an artist. There are pleasures in parsimony, in squeezing the last drop of performance out of clever algorithms and tight coding. The games, puzzles, and challenges of problems from international programming competitions are a great way to experience these pleasures while improving your algorithmic and coding skills. This book contains over 100 problems that have appeared in previous programming contests, along with discussions of the theory and ideas necessary to attack them. Instant online grading for all of these problems is available from two WWW robot judging sites. Combining this book with a judge gives an exciting new way to challenge and improve your programming skills. This book can be used for self-study, for teaching innovative courses in algorithms and programming, and in training for international competition. The problems in this book have been selected from over 1,000 programming problems at the Universidad de Valladolid online judge. The judge has ruled on well over one million submissions from 27,000 registered users around the world to date. We have taken only the best of the best, the most fun, exciting, and interesting problems available.

array reduction 3 hackerrank solution: Approximation Algorithms Vijay V. Vazirani, 2013-03-14 Covering the basic techniques used in the latest research work, the author consolidates progress made so far, including some very recent and promising results, and conveys the beauty and excitement of work in the field. He gives clear, lucid explanations of key results and ideas, with intuitive proofs, and provides critical examples and numerous illustrations to help elucidate the algorithms. Many of the results presented have been simplified and new insights provided. Of interest to theoretical computer scientists, operations researchers, and discrete mathematicians.

array reduction 3 hackerrank solution: How Smart Machines Think Sean Gerrish, 2018-10-30 Everything you've always wanted to know about self-driving cars, Netflix recommendations, IBM's Watson, and video game-playing computer programs. The future is here: Self-driving cars are on the streets, an algorithm gives you movie and TV recommendations, IBM's Watson triumphed on Jeopardy over puny human brains, computer programs can be trained to play Atari games. But how do all these things work? In this book, Sean Gerrish offers an engaging and accessible overview of the breakthroughs in artificial intelligence and machine learning that have made today's machines so smart. Gerrish outlines some of the key ideas that enable intelligent machines to perceive and interact with the world. He describes the software architecture that allows self-driving cars to stay on the road and to navigate crowded urban environments; the million-dollar Netflix competition for a better recommendation engine (which had an unexpected ending); and how programmers trained computers to perform certain behaviors by offering them treats, as if they were training a dog. He explains how artificial neural networks enable computers to perceive the world—and to play Atari video games better than humans. He explains Watson's famous victory on Jeopardy, and he looks at how computers play games, describing AlphaGo and Deep Blue, which beat reigning world champions at the strategy games of Go and chess. Computers have not yet mastered everything, however; Gerrish outlines the difficulties in creating intelligent agents that can successfully play video games like StarCraft that have evaded solution—at least for now. Gerrish weaves the stories behind these breakthroughs into the narrative, introducing readers to many of the researchers involved, and keeping technical details to a minimum. Science and technology buffs will find this book an essential guide to a future in which machines can outsmart people.

array reduction 3 hackerrank solution: Quant Job Interview Questions and Answers Mark Joshi, Nick Denson, Nicholas Denson, Andrew Downes, 2013 The quant job market has never been tougher. Extensive preparation is essential. Expanding on the successful first edition, this second edition has been updated to reflect the latest questions asked. It now provides over 300 interview questions taken from actual interviews in the City and Wall Street. Each question comes with a full detailed solution, discussion of what the interviewer is seeking and possible follow-up questions. Topics covered include option pricing, probability, mathematics, numerical algorithms and C++, as well as a discussion of the interview process and the non-technical interview. All three authors have

worked as quants and they have done many interviews from both sides of the desk. Mark Joshi has written many papers and books including the very successful introductory textbook, The Concepts and Practice of Mathematical Finance.

array reduction 3 hackerrank solution: *Mastering Oracle PL/SQL* Christopher Beck, Joel Kallman, Chaim Katz, David C. Knox, Connor McDonald, 2008-01-01 If you have mastered the fundamentals of the PL/SQL language and are now looking for an in-depth, practical guide to solving real problems with PL/SQL stored procedures, then this is the book for you.

array reduction 3 hackerrank solution: Algorithmic Puzzles Anany Levitin, Maria Levitin, 2011-10-14 Algorithmic puzzles are puzzles involving well-defined procedures for solving problems. This book will provide an enjoyable and accessible introduction to algorithmic puzzles that will develop the reader's algorithmic thinking. The first part of this book is a tutorial on algorithm design strategies and analysis techniques. Algorithm design strategies — exhaustive search, backtracking, divide-and-conquer and a few others — are general approaches to designing step-by-step instructions for solving problems. Analysis techniques are methods for investigating such procedures to answer questions about the ultimate result of the procedure or how many steps are executed before the procedure stops. The discussion is an elementary level, with puzzle examples, and requires neither programming nor mathematics beyond a secondary school level. Thus, the tutorial provides a gentle and entertaining introduction to main ideas in high-level algorithmic problem solving. The second and main part of the book contains 150 puzzles, from centuries-old classics to newcomers often asked during job interviews at computing, engineering, and financial companies. The puzzles are divided into three groups by their difficulty levels. The first fifty puzzles in the Easier Puzzles section require only middle school mathematics. The sixty puzzle of average difficulty and forty harder puzzles require just high school mathematics plus a few topics such as binary numbers and simple recurrences, which are reviewed in the tutorial. All the puzzles are provided with hints, detailed solutions, and brief comments. The comments deal with the puzzle origins and design or analysis techniques used in the solution. The book should be of interest to puzzle lovers, students and teachers of algorithm courses, and persons expecting to be given puzzles during job interviews.

array reduction 3 hackerrank solution: Mastering Algorithms with C Kyle Loudon, 1999 Implementations, as well as interesting, real-world examples of each data structure and algorithm, are shown in the text. Full source code appears on the accompanying disk.

array reduction 3 hackerrank solution: Measuring the Digital Transformation A Roadmap for the Future OECD, 2019-03-11 Measuring the Digital Transformation: A Roadmap for the Future provides new insights into the state of the digital transformation by mapping indicators across a range of areas – from education and innovation, to trade and economic and social outcomes – against current digital policy issues, as presented in Going Digital: Shaping Policies, Improving Lives.

array reduction 3 hackerrank solution: *Computer Science* Robert Sedgewick, Kevin Wayne, 2016-06-17 Named a Notable Book in the 21st Annual Best of Computing list by the ACM! Robert Sedgewick and Kevin Wayne’s Computer Science: An Interdisciplinary Approach is the ideal modern introduction to computer science with Java programming for both students and professionals. Taking a broad, applications-based approach, Sedgewick and Wayne teach through important examples from science, mathematics, engineering, finance, and commercial computing. The book demystifies computation, explains its intellectual underpinnings, and covers the essential elements of programming and computational problem solving in today’s environments. The authors begin by introducing basic programming elements such as variables, conditionals, loops, arrays, and I/O. Next, they turn to functions, introducing key modular programming concepts, including components and reuse. They present a modern introduction to object-oriented programming, covering current programming paradigms and approaches to data abstraction. Building on this foundation, Sedgewick and Wayne widen their focus to the broader discipline of computer science. They introduce classical sorting and searching algorithms, fundamental data structures and their application, and scientific

techniques for assessing an implementation’s performance. Using abstract models, readers learn to answer basic questions about computation, gaining insight for practical application. Finally, the authors show how machine architecture links the theory of computing to real computers, and to the field’s history and evolution. For each concept, the authors present all the information readers need to build confidence, together with examples that solve intriguing problems. Each chapter contains question-and-answer sections, self-study drills, and challenging problems that demand creative solutions. Companion web site (introcs.cs.princeton.edu/java) contains Extensive supplementary information, including suggested approaches to programming assignments, checklists, and FAQs Graphics and sound libraries Links to program code and test data Solutions to selected exercises Chapter summaries Detailed instructions for installing a Java programming environment Detailed problem sets and projects Companion 20-part series of video lectures is available at informit.com/title/9780134493831

array reduction 3 hackerrank solution: Programming Interviews Exposed John Mongan, Noah Suojanen Kindler, Eric Giguère, 2011-08-10 The pressure is on during the interview process but with the right preparation, you can walk away with your dream job. This classic book uncovers what interviews are really like at America's top software and computer companies and provides you with the tools to succeed in any situation. The authors take you step-by-step through new problems and complex brainteasers they were asked during recent technical interviews. 50 interview scenarios are presented along with in-depth analysis of the possible solutions. The problem-solving process is clearly illustrated so you'll be able to easily apply what you've learned during crunch time. You'll also find expert tips on what questions to ask, how to approach a problem, and how to recover if you become stuck. All of this will help you ace the interview and get the job you want. What you will learn from this book Tips for effectively completing the job application Ways to prepare for the entire programming interview process How to find the kind of programming job that fits you best Strategies for choosing a solution and what your approach says about you How to improve your interviewing skills so that you can respond to any question or situation Techniques for solving knowledge-based problems, logic puzzles, and programming problems Who this book is for This book is for programmers and developers applying for jobs in the software industry or in IT departments of major corporations. Wrox Beginning guides are crafted to make learning programming languages and technologies easier than you think, providing a structured, tutorial format that will guide you through all the techniques involved.

array reduction 3 hackerrank solution: Advances in Smart Vehicular Technology, Transportation, Communication and Applications Yong Zhao, Tsu-Yang Wu, Tang-Hsien Chang, Jeng-Shyang Pan, Lakhmi C. Jain, 2018-11-30 This book highlights papers presented at the Second International Conference on Smart Vehicular Technology, Transportation, Communication and Applications (VTCA 2018), which was held at Mount Emei, Sichuan Province, China from 25 to 28 October 2018. The conference was co-sponsored by Springer, Southwest Jiaotong University, Fujian University of Technology, Chang’an University, Shandong University of Science and Technology, Fujian Provincial Key Lab of Big Data Mining and Applications, and the National Demonstration Center for Experimental Electronic Information and Electrical Technology Education (Fujian University of Technology). The conference was intended as an international forum for researchers and professionals engaged in all areas of smart vehicular technology, vehicular transportation, vehicular communication, and applications.

array reduction 3 hackerrank solution: Programming Collective Intelligence Toby Segaran, 2007-08-16 Want to tap the power behind search rankings, product recommendations, social bookmarking, and online matchmaking? This fascinating book demonstrates how you can build Web 2.0 applications to mine the enormous amount of data created by people on the Internet. With the sophisticated algorithms in this book, you can write smart programs to access interesting datasets from other web sites, collect data from users of your own applications, and analyze and understand the data once you've found it. Programming Collective Intelligence takes you into the world of machine learning and statistics, and explains how to draw conclusions about user

experience, marketing, personal tastes, and human behavior in general -- all from information that you and others collect every day. Each algorithm is described clearly and concisely with code that can immediately be used on your web site, blog, Wiki, or specialized application. This book explains: Collaborative filtering techniques that enable online retailers to recommend products or media Methods of clustering to detect groups of similar items in a large dataset Search engine features -- crawlers, indexers, query engines, and the PageRank algorithm Optimization algorithms that search millions of possible solutions to a problem and choose the best one Bayesian filtering, used in spam filters for classifying documents based on word types and other features Using decision trees not only to make predictions, but to model the way decisions are made Predicting numerical values rather than classifications to build price models Support vector machines to match people in online dating sites Non-negative matrix factorization to find the independent features in a dataset Evolving intelligence for problem solving -- how a computer develops its skill by improving its own code the more it plays a game Each chapter includes exercises for extending the algorithms to make them more powerful. Go beyond simple database-backed applications and put the wealth of Internet data to work for you. Bravo! I cannot think of a better way for a developer to first learn these algorithms and methods, nor can I think of a better way for me (an old AI dog) to reinvigorate my knowledge of the details. -- Dan Russell, Google Toby's book does a great job of breaking down the complex subject matter of machine-learning algorithms into practical, easy-to-understand examples that can be directly applied to analysis of social interaction across the Web today. If I had this book two years ago, it would have saved precious time going down some fruitless paths. -- Tim Wolters, CTO, Collective Intellect

array reduction 3 hackerrank solution: The Golden Ticket Lance Fortnow, 2017-02-28 The computer science problem whose solution could transform life as we know it The P-NP problem is the most important open problem in computer science, if not all of mathematics. Simply stated, it asks whether every problem whose solution can be quickly checked by computer can also be quickly solved by computer. The Golden Ticket provides a nontechnical introduction to P-NP, its rich history, and its algorithmic implications for everything we do with computers and beyond. Lance Fortnow traces the history and development of P-NP, giving examples from a variety of disciplines, including economics, physics, and biology. He explores problems that capture the full difficulty of the P-NP dilemma, from discovering the shortest route through all the rides at Disney World to finding large groups of friends on Facebook. The Golden Ticket explores what we truly can and cannot achieve computationally, describing the benefits and unexpected challenges of this compelling problem.

array reduction 3 hackerrank solution: Applied Asymptotic Analysis Peter David Miller, 2006 This book is a survey of asymptotic methods set in the current applied research context of wave propagation. It stresses rigorous analysis in addition to formal manipulations. Asymptotic expansions developed in the text are justified rigorously, and students are shown how to obtain solid error estimates for asymptotic formulae. The book relates examples and exercises to subjects of current research interest, such as the problem of locating the zeros of Taylor polynomials of entirenonvanishing functions and the problem of counting integer lattice points in subsets of the plane with various geometrical properties of the boundary. The book is intended for a beginning graduate course on asymptotic analysis in applied mathematics and is aimed at students of pure and appliedmathematics as well as science and engineering. The basic prerequisite is a background in differential equations, linear algebra, advanced calculus, and complex variables at the level of introductory undergraduate courses on these subjects. The book is ideally suited to the needs of a graduate student who, on the one hand, wants to learn basic applied mathematics, and on the other, wants to understand what is needed to make the various arguments rigorous. Down here in the Village, this is knownas the Courant point of view!! --Percy Deift, Courant Institute, New York Peter D. Miller is an associate professor of mathematics at the University of Michigan at Ann Arbor. He earned a Ph.D. in Applied Mathematics from the University of Arizona and has held positions at the Australian NationalUniversity (Canberra) and Monash University (Melbourne). His current research interests lie in singular limits for integrable systems.

array reduction 3 hackerrank solution: *Data Structures Using C* Reema Thareja, 2014 This second edition of Data Structures Using C has been developed to provide a comprehensive and consistent coverage of both the abstract concepts of data structures as well as the implementation of these concepts using C language. It begins with a thorough overview of the concepts of C programming followed by introduction of different data structures and methods to analyse the complexity of different algorithms. It then connects these concepts and applies them to the study of various data structures such as arrays, strings, linked lists, stacks, queues, trees, heaps, and graphs. The book utilizes a systematic approach wherein the design of each of the data structures is followed by algorithms of different operations that can be performed on them, and the analysis of these algorithms in terms of their running times. Each chapter includes a variety of end-chapter exercises in the form of MCQs with answers, review questions, and programming exercises to help readers test their knowledge.

array reduction 3 hackerrank solution: *Algorithms* Sanjoy Dasgupta, Christos H. Papadimitriou, Umesh Virkumar Vazirani, 2006 This text, extensively class-tested over a decade at UC Berkeley and UC San Diego, explains the fundamentals of algorithms in a story line that makes the material enjoyable and easy to digest. Emphasis is placed on understanding the crisp mathematical idea behind each algorithm, in a manner that is intuitive and rigorous without being unduly formal. Features include:The use of boxes to strengthen the narrative: pieces that provide historical context, descriptions of how the algorithms are used in practice, and excursions for the mathematically sophisticated. Carefully chosen advanced topics that can be skipped in a standard one-semester course but can be covered in an advanced algorithms course or in a more leisurely two-semester sequence.An accessible treatment of linear programming introduces students to one of the greatest achievements in algorithms. An optional chapter on the quantum algorithm for factoring provides a unique peephole into this exciting topic. In addition to the text DasGupta also offers a Solutions Manual which is available on the Online Learning Center.Algorithms is an outstanding undergraduate text equally informed by the historical roots and contemporary applications of its subject. Like a captivating novel it is a joy to read. Tim Roughgarden Stanford University

array reduction 3 hackerrank solution: Data Structures and Algorithms Made Easy CareerMonk Publications, Narasimha Karumanchi, 2008-05-05 Data Structures And Algorithms Made Easy: Data Structure And Algorithmic Puzzles is a book that offers solutions to complex data structures and algorithms. There are multiple solutions for each problem and the book is coded in C/C++, it comes handy as an interview and exam guide for computer...

array reduction 3 hackerrank solution: Patently Mathematical Jeff Suzuki, 2018-12-14 Uncovers the surprising ways math shapes our lives—from whom we date to what we learn. How do dating sites match compatible partners? What do cell phones and sea coasts have in common? And why do computer scientists keep ant colonies? Jeff Suzuki answers these questions and more in Patently Mathematical, which explores the mathematics behind some of the key inventions that have changed our world. In recent years, patents based on mathematics have been issued by the thousands—from search engines and image recognition technology to educational software and LEGO designs. Suzuki delves into the details of cutting-edge devices, programs, and products to show how even the simplest mathematical principles can be turned into patentable ideas worth billions of dollars. Readers will discover • whether secure credit cards are really secure • how improved data compression made streaming video services like Netflix a hit • the mathematics behind self-correcting golf balls • why Google is such an effective and popular search engine • how eHarmony and Match.com find the perfect partner for those seeking a mate • and much more! A gifted writer who combines quirky historical anecdotes with relatable, everyday examples, Suzuki makes math interesting for everyone who likes to ponder the world of numerical relationships. Praise for Jeff Suzuki's Constitutional Calculus Presents an entertaining and insightful approach to the mathematics that underlies the American system of government. The book is neatly organized, breaking down the United States Constitution by article, section, and amendment. Within each piece, Suzuki reviews the mathematical principles that went into the underlying

framework.—Mathematical Reviews A breath of fresh air. . . . A reaffirmation that mathematics should be used more often to make general public policy.—MAA Reviews

array reduction 3 hackerrank solution: Head First PHP & MySQL Lynn Beighley, Michael Morrison, 2009 With this book, Web designers who usually turn out static Websites with HTML and CSS can make the leap to the next level of Web development--full-fledged, dynamic, database-driven Websites using PHP and SQL.

array reduction 3 hackerrank solution: *Data Structures and Algorithms in C++* Michael T. Goodrich, Roberto Tamassia, David M. Mount, 2011-02-22 An updated, innovative approach to data structures and algorithms Written by an author team of experts in their fields, this authoritative guide demystifies even the most difficult mathematical concepts so that you can gain a clear understanding of data structures and algorithms in C++. The unparalleled author team incorporates the object-oriented design paradigm using C++ as the implementation language, while also providing intuition and analysis of fundamental algorithms. Offers a unique multimedia format for learning the fundamentals of data structures and algorithms Allows you to visualize key analytic concepts, learn about the most recent insights in the field, and do data structure design Provides clear approaches for developing programs Features a clear, easy-to-understand writing style that breaks down even the most difficult mathematical concepts Building on the success of the first edition, this new version offers you an innovative approach to fundamental data structures and algorithms.

array reduction 3 hackerrank solution: *Advances in Decision Sciences, Image Processing, Security and Computer Vision* Suresh Chandra Satapathy, K. Srujan Raju, K. Shyamala, D. Rama Krishna, Margarita N. Favorskaya, 2019-07-12 This book constitutes the proceedings of the First International Conference on Emerging Trends in Engineering (ICETE), held at University College of Engineering and organised by the Alumni Association, University College of Engineering, Osmania University, in Hyderabad, India on 22–23 March 2019. The proceedings of the ICETE are published in three volumes, covering seven areas: Biomedical, Civil, Computer Science, Electrical & Electronics, Electronics & Communication, Mechanical, and Mining Engineering. The 215 peer-reviewed papers from around the globe present the latest state-of-the-art research, and are useful to postgraduate students, researchers, academics and industry engineers working in the respective fields. Volume 1 presents papers on the theme “Advances in Decision Sciences, Image Processing, Security and Computer Vision – International Conference on Emerging Trends in Engineering (ICETE)”. It includes state-of-the-art technical contributions in the area of biomedical and computer science engineering, discussing sustainable developments in the field, such as instrumentation and innovation, signal and image processing, Internet of Things, cryptography and network security, data mining and machine learning.

array reduction 3 hackerrank solution: The Financial Mathematics of Market Liquidity Olivier Gueant, 2016-03-30 This book is among the first to present the mathematical models most commonly used to solve optimal execution problems and market making problems in finance. The Financial Mathematics of Market Liquidity: From Optimal Execution to Market Making presents a general modeling framework for optimal execution problems-inspired from the Almgren-Chriss app

array reduction 3 hackerrank solution: How Can Self-learners Learn Programming in the Most Efficient Way? A Pragmatic Approach Sebastien Phlix, 2016-12-13 Master's Thesis from the year 2016 in the subject Computer Science - Programming, grade: 20/20, Ecole des hautes etudes commerciales de Paris (HEC Entrepreneurs), language: English, abstract: This paper provides a structured approach for self-learning programming for free on the internet. Its recommendations are based on a review of the existing academic literature which is complemented by the analysis of numerous contributions by software developers, self-learners, and teachers of programming. Additionally, it incorporates effective learning techniques derived from psychological research. Its intended readers are primarily entrepreneurs and 'startup people' who are driven to build new businesses with code, although the proposed approach is also transferable to other domains and audiences. The single most important factor for succeeding in learning programming

has been found to be of human nature: learner motivation and persistence. While most beginners and the majority of academic contributions focus mostly on technical aspects such as which language to learn first, or which learning resources to use, this paper analyzes the learning process itself. Learning programming is thus divided into three main steps: First, I highlight the importance of setting a strong learning goal for motivation, and provide a big-picture overview of what 'learning programming' encompasses to structure the approach. Second, I provide learners with recommendations as to which language to learn first - there is no one 'best' choice - as well as how and where to find effective learning resources. Lastly, the paper concludes with tips for optimizing the learning process by introducing effective learning techniques, highlighting the importance of programming practice, and collecting additional advice from programmers and self-learners.

array reduction 3 hackerrank solution: Politics and the English Language George Orwell, 2021-01-01 George Orwell set out ‘to make political writing into an art’, and to a wide extent this aim shaped the future of English literature – his descriptions of authoritarian regimes helped to form a new vocabulary that is fundamental to understanding totalitarianism. While 1984 and Animal Farm are amongst the most popular classic novels in the English language, this new series of Orwell’s essays seeks to bring a wider selection of his writing on politics and literature to a new readership. In Politics and the English Language, the second in the Orwell’s Essays series, Orwell takes aim at the language used in politics, which, he says, ‘is designed to make lies sound truthful and murder respectable, and to give an appearance of solidity to pure wind’. In an age where the language used in politics is constantly under the microscope, Orwell’s Politics and the English Language is just as relevant today, and gives the reader a vital understanding of the tactics at play. 'A writer who can – and must – be rediscovered with every age.' — Irish Times

array reduction 3 hackerrank solution: Grokking the System Design Interview Design Gurus, 2021-12-18 This book (also available online at www.designgurus.org) by Design Gurus has helped 60k+ readers to crack their system design interview (SDI). System design questions have become a standard part of the software engineering interview process. These interviews determine your ability to work with complex systems and the position and salary you will be offered by the interviewing company. Unfortunately, SDI is difficult for most engineers, partly because they lack experience developing large-scale systems and partly because SDIs are unstructured in nature. Even engineers who've some experience building such systems aren't comfortable with these interviews, mainly due to the open-ended nature of design problems that don't have a standard answer. This book is a comprehensive guide to master SDIs. It was created by hiring managers who have worked for Google, Facebook, Microsoft, and Amazon. The book contains a carefully chosen set of questions that have been repeatedly asked at top companies. What's inside? This book is divided into two parts. The first part includes a step-by-step guide on how to answer a system design question in an interview, followed by famous system design case studies. The second part of the book includes a glossary of system design concepts. Table of Contents First Part: System Design Interviews: A step-by-step guide. Designing a URL Shortening service like TinyURL. Designing Pastebin. Designing Instagram. Designing Dropbox. Designing Facebook Messenger. Designing Twitter. Designing YouTube or Netflix. Designing Typeahead Suggestion. Designing an API Rate Limiter. Designing Twitter Search. Designing a Web Crawler. Designing Facebook's Newsfeed. Designing Yelp or Nearby Friends. Designing Uber backend. Designing Ticketmaster. Second Part: Key Characteristics of Distributed Systems. Load Balancing. Caching. Data Partitioning. Indexes. Proxies. Redundancy and Replication. SQL vs. NoSQL. CAP Theorem. PACELC Theorem. Consistent Hashing. Long-Polling vs. WebSockets vs. Server-Sent Events. Bloom Filters. Quorum. Leader and Follower. Heartbeat. Checksum. About the Authors Designed Gurus is a platform that offers online courses to help software engineers prepare for coding and system design interviews. Learn more about our courses at www.designgurus.org.