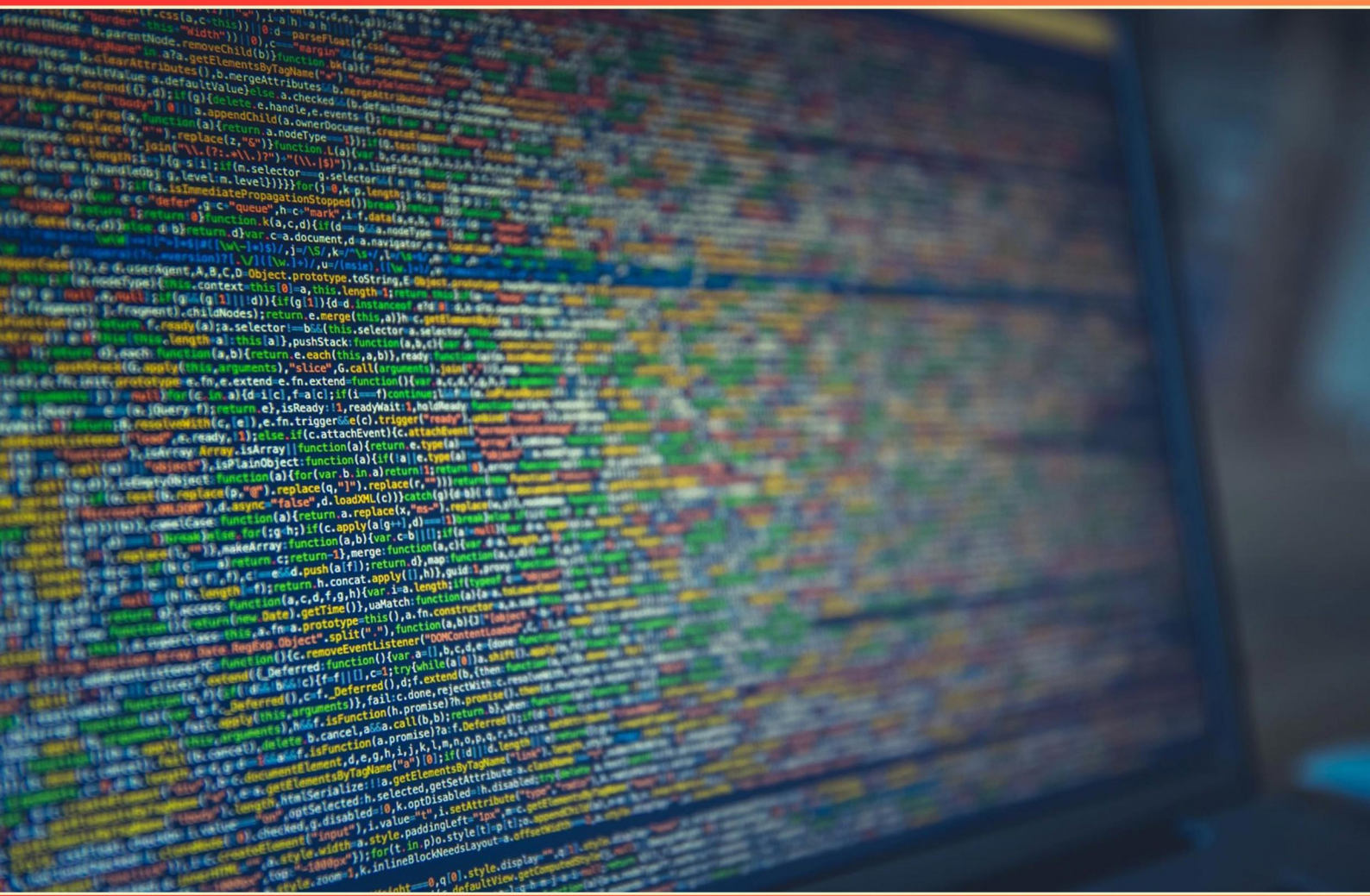
WESTERN GOVERNORS UNIVERSITY (WGU) C839V5 / D334 ALGORITHMS PRACTICE TEST (SAMPLE)

STUDY GUIDE



SAMPLE STUDY GUIDE WITH LIMITED QUESTIONS

VISIT US ONLINE FOR THE FULL VERSION STUDY GUIDE

https://wgu-d334algorithms.examzify.com

Copyright © 2026 by Examzify - A Kaluba Technologies Inc. product.

ALL RIGHTS RESERVED.

No part of this book may be reproduced or transferred in any form or by any means, graphic, electronic, or mechanical, including photocopying, recording, web distribution, taping, or by any information storage retrieval system, without the written permission of the author.

Notice: Examzify makes every reasonable effort to obtain accurate, complete, and timely information about this product from reliable sources.

SAMPLE

Table of Contents

Copyright 1

Table of Contents 2

Introduction 3

How to Use This Guide 4

Questions 5

Answers 8

Explanations 10

Next Steps 16

SAMPLE

Introduction

Preparing for a certification exam can feel overwhelming, but with the right tools, it becomes an opportunity to build confidence, sharpen your skills, and move one step closer to your goals. At Examzify, we believe that effective exam preparation isn't just about memorization, it's about understanding the material, identifying knowledge gaps, and building the test-taking strategies that lead to success.

This guide was designed to help you do exactly that.

Whether you're preparing for a licensing exam, professional certification, or entry-level qualification, this book offers structured practice to reinforce key concepts. You'll find a wide range of multiple-choice questions, each followed by clear explanations to help you understand not just the right answer, but why it's correct.

The content in this guide is based on real-world exam objectives and aligned with the types of questions and topics commonly found on official tests. It's ideal for learners who want to:

- Practice answering questions under realistic conditions,
- Improve accuracy and speed,
- Review explanations to strengthen weak areas, and
- Approach the exam with greater confidence.

We recommend using this book not as a stand-alone study tool, but alongside other resources like flashcards, textbooks, or hands-on training. For best results, we recommend working through each question, reflecting on the explanation provided, and revisiting the topics that challenge you most.

Remember: successful test preparation isn't about getting every question right the first time, it's about learning from your mistakes and improving over time. Stay focused, trust the process, and know that every page you turn brings you closer to success.

Let's begin.

SAMPLE

How to Use This Guide

This guide is designed to help you study more effectively and approach your exam with confidence. Whether you're reviewing for the first time or doing a final refresh, here's how to get the most out of your Examzify study guide:

1. Start with a Diagnostic Review

Skim through the questions to get a sense of what you know and what you need to focus on. Your goal is to identify knowledge gaps early.

2. Study in Short, Focused Sessions

Break your study time into manageable blocks (e.g. 30 – 45 minutes). Review a handful of questions, reflect on the explanations.

3. Learn from the Explanations

After answering a question, always read the explanation, even if you got it right. It reinforces key points, corrects misunderstandings, and teaches subtle distinctions between similar answers.

4. Track Your Progress

Use bookmarks or notes (if reading digitally) to mark difficult questions. Revisit these regularly and track improvements over time.

5. Simulate the Real Exam

Once you're comfortable, try taking a full set of questions without pausing. Set a timer and simulate test-day conditions to build confidence and time management skills.

6. Repeat and Review

Don't just study once, repetition builds retention. Re-attempt questions after a few days and revisit explanations to reinforce learning. Pair this guide with other Examzify tools like flashcards, and digital practice tests to strengthen your preparation across formats.

There's no single right way to study, but consistent, thoughtful effort always wins. Use this guide flexibly, adapt the tips above to fit your pace and learning style. You've got this!

SAMPLE

Questions

SAMPLE

1. What characteristic does a stack have in comparison to other data structures?

A. Elements can be accessed in any order

B. Elements are added and removed in a FIFO manner

C. Elements are processed in a LIFO manner

D. Elements can only be viewed, not modified

2. What distinguishes a heuristic algorithm from a brute force algorithm?

A. A heuristic algorithm uses exhaustive search

B. A brute force algorithm is faster

C. A brute force algorithm uses exhaustive search for a solution

D. A heuristic algorithm always guarantees the optimal solution

3. What does graph isomorphism refer to?

A. Two graphs that have different numbers of vertices.

B. Two graphs that can be transformed into one another by renaming vertices.

C. Two graphs that share identical edge weights.

D. Two graphs that are completely unrelated.

4. What output length is typical for the MD5 hash function?

A. 160 bits

B. 128 bits

C. 256 bits

D. 512 bits

5. What is the key feature of the bubble sort algorithm?

A. It uses recursive calls

B. It utilizes randomization to sort

C. It repeatedly compares and swaps adjacent elements

D. It constructs a sorted list from an unsorted list

6. What key sizes can be used with AES encryption?

A. 64 bits or 128 bits

B. 128 bits, 192 bits, or 256 bits

C. 128 bits or 256 bits

D. 128 bits, 256 bits, or 512 bits

SAMPLE

7. How does a recursive algorithm operate?

A. By using a loop to repeat actions

B. By storing and reusing results from previous operations

C. By calling itself within its definition

D. By dividing input into equal parts only

8. What is a linked list?

A. A data structure consisting of a sequence of elements, where each element points to the next

B. A linear data structure that stores data elements in a contiguous manner

C. A hierarchical data structure used to represent relationships

D. A fixed-size array that allows for efficient random access

9. What is the Traveling Salesperson Problem (TSP)?

A. A problem that selects a maximum number of activities

B. A problem that minimizes the cost of goods

C. An optimization problem requiring visits to all cities

D. A search algorithm for finding the best path

10. Which of the following is NOT a characteristic of Merge Sort?

A. It sorts elements by merging sorted halves

B. It requires space for temporary arrays during processing

C. It operates efficiently on large, unsorted lists

D. It performs better than Quick Sort in all cases

SAMPLE

Answers

SAMPLE

1. C
2. C
3. B
4. B
5. C
6. B
7. C
8. A
9. C
10. D

SAMPLE

Explanations

SAMPLE

1. What characteristic does a stack have in comparison to other data structures?

A. Elements can be accessed in any order

B. Elements are added and removed in a FIFO manner

C. Elements are processed in a LIFO manner

D. Elements can only be viewed, not modified

A stack is a special type of data structure that follows the Last In, First Out (LIFO) principle. This means that the most recently added element is the one that will be the first to be removed when an element is accessed. The characteristic of processing in a LIFO manner is fundamental to how stacks operate, differentiating them from other structures like queues, which use a First In, First Out (FIFO) approach. In a stack, operations such as push (adding an element) and pop (removing an element) are performed only at one end, known as the top of the stack. This access pattern enforces the LIFO behavior, making the stack particularly useful for scenarios such as backtracking algorithms, function call management in programming, and parsing expressions. Other characteristics of stacks may include the fact that not all elements can be accessed randomly, and they do not support operations that allow elements to be added or removed from the middle of the structure. Consequently, the LIFO processing is the defining characteristic that sets stacks apart from other data structures.

2. What distinguishes a heuristic algorithm from a brute force algorithm?

A. A heuristic algorithm uses exhaustive search

B. A brute force algorithm is faster

C. A brute force algorithm uses exhaustive search for a solution

D. A heuristic algorithm always guarantees the optimal solution

A brute force algorithm is characterized by its method of problem-solving through exhaustive search, meaning it explores all possible solutions until it finds the correct one. This can be effective for smaller problems or when the solution space is limited, as it guarantees finding the optimal solution as it evaluates every possible option. In contrast, heuristic algorithms offer a different approach. They are designed to find solutions more quickly and efficiently, particularly when dealing with larger or more complex problems where exhaustive searching would be computationally infeasible. Heuristic methods typically provide good enough solutions within a reasonable time frame, but they do not guarantee optimality, which is a key distinction from brute force methods. Additionally, while brute force algorithms systematically check all possibilities, heuristic algorithms rely on experience, rules of thumb, or approximative techniques to navigate the solution space more intelligently. This strategic approach often results in faster performance, but at the potential cost of not reaching the best possible solution every time. Understanding these differences is crucial in deciding which algorithm to use based on the problem at hand, the size of the input, and the importance of finding an optimal versus a satisfactory solution.

SAMPLE

3. What does graph isomorphism refer to?

A. Two graphs that have different numbers of vertices.

B. Two graphs that can be transformed into one another by renaming vertices.

C. Two graphs that share identical edge weights.

D. Two graphs that are completely unrelated.

Graph isomorphism refers to the relationship between two graphs that can be transformed into one another through a relabeling of their vertices. This means that there exists a one-to-one correspondence between the vertices of the two graphs such that if two vertices are connected by an edge in one graph, their corresponding vertices in the other graph are also connected by an edge. Essentially, isomorphic graphs have the same structure or connectivity, even if their vertex names differ. This concept is crucial in many areas of graph theory and computer science, as it helps in identifying when two graphs are fundamentally the same despite possibly being labeled differently or appearing distinct at first glance. Understanding graph isomorphism is essential for solving various problems related to network analysis, data representation, and pattern recognition.

4. What output length is typical for the MD5 hash function?

A. 160 bits

B. 128 bits

C. 256 bits

D. 512 bits

The MD5 hash function produces a fixed output length of 128 bits, which is equivalent to 32 hexadecimal characters. This characteristic is essential for achieving consistent hash values regardless of the input size. Understanding the output length is crucial when implementing cryptographic functions, as it determines the level of security and the potential for hash collisions. In contrast, other hash functions like SHA-1 produce a 160-bit output, while SHA-256 generates a 256-bit hash, and SHA-512 offers a 512-bit output. These different output lengths reflect the trade-offs between performance and security; however, for MD5 specifically, the definitive output is 128 bits. This makes it important for anyone using MD5 to be aware of its limitations, as it has been found vulnerable to various forms of attacks, leading to its decline in use for security-sensitive applications.

5. What is the key feature of the bubble sort algorithm?

A. It uses recursive calls

B. It utilizes randomization to sort

C. It repeatedly compares and swaps adjacent elements

D. It constructs a sorted list from an unsorted list

The bubble sort algorithm is characterized by its method of sorting elements through a series of repeated comparisons and swaps of adjacent elements. In each pass through the list, the algorithm compares each pair of adjacent items and swaps them if they are in the wrong order. This process continues iteratively, resulting in the largest unsorted element "bubbling up" to its correct position at the end of the list with each complete pass. This mechanism is what gives bubble sort its name and is crucial to its operation. Each complete iteration through the list ensures that the adjacent comparisons lead to a gradually sorted arrangement, highlighting the simplicity and straightforward nature of this sorting method. Other options such as using recursive calls or randomization do not describe bubble sort accurately, as bubble sort specifically relies on a straightforward iterative approach. While constructing a sorted list is a goal of many sorting algorithms, bubble sort uniquely does this through its adjacent comparison and swap method, leading to its defining feature being rightly identified in the provided answer.

SAMPLE

6. What key sizes can be used with AES encryption?

A. 64 bits or 128 bits

B. 128 bits, 192 bits, or 256 bits

C. 128 bits or 256 bits

D. 128 bits, 256 bits, or 512 bits

The key sizes that can be used with AES (Advanced Encryption Standard) encryption are 128 bits, 192 bits, and 256 bits. AES is a symmetric encryption algorithm established by NIST, and it supports these three key lengths, which provide different levels of security. Using a longer key length, such as 192 bits or 256 bits, increases the security of the encryption by making it more difficult for an attacker to perform a brute-force attack. The choice of key length allows an organization to balance the need for security against performance considerations, as longer keys require more computational power for both encryption and decryption. Key sizes of 64 bits or 512 bits are not applicable to AES. A 64-bit key size is more characteristic of older encryption standards, like DES (Data Encryption Standard), while a 512-bit key is beyond what AES supports and generally involves other cryptographic systems that are not AES. Thus, the only valid key sizes for AES are indeed 128 bits, 192 bits, and 256 bits.

7. How does a recursive algorithm operate?

A. By using a loop to repeat actions

B. By storing and reusing results from previous operations

C. By calling itself within its definition

D. By dividing input into equal parts only

A recursive algorithm operates by calling itself within its definition, allowing it to solve larger problems by breaking them down into smaller, more manageable subproblems. This self-referential approach enables the algorithm to handle tasks in a structured, often elegant manner. For example, consider a factorial function which calculates the product of integers from 1 to n. A recursive implementation would multiply n by the factorial of (n-1), effectively reducing the problem size each time the function calls itself until it reaches a base case, such as factorial(1) which simply returns 1. This mechanism of self-calling is what distinguishes recursive algorithms from iterative ones, which rely on loops. The other options focus on different mechanisms that aren't characteristic of recursive algorithms. While loops are a feature of iterative approaches, caching results refers more to dynamic programming than recursion directly, and dividing input into equal parts describes approaches like divide-and-conquer rather than the recursive nature itself. Thus, the essence of recursion lies in this ability to call itself, facilitating a divide-and-conquer strategy or simply reducing the problem size directly within its structure.

SAMPLE

8. What is a linked list?

A. A data structure consisting of a sequence of elements, where each element points to the next

B. A linear data structure that stores data elements in a contiguous manner

C. A hierarchical data structure used to represent relationships

D. A fixed-size array that allows for efficient random access

A linked list is a data structure consisting of a sequence of elements, where each element points to the next. This organization allows for dynamic memory allocation, meaning that the size can change during program execution. Each element in a linked list, often referred to as a node, includes data and a reference (or pointer) to the next node in the sequence. This structure enables efficient insertion and deletion of elements, as nodes can be easily added or removed by adjusting the pointers, without needing to shift other elements around as would be necessary in an array. Additionally, linked lists can easily grow or shrink in size, accommodating varying amounts of data without requiring a predefined array size. In contrast, other options describe different data structures. For instance, linear structures that store elements contiguously refer to arrays, which lack the flexibility of linked lists. Hierarchical structures describe trees, which organize data in a branching manner rather than a sequential one. Fixed-size arrays limit the dynamic aspect of data organization and can incur inefficiencies in handling variable-sized datasets. Therefore, the description of a linked list is most accurately encapsulated in the first choice.

9. What is the Traveling Salesperson Problem (TSP)?

A. A problem that selects a maximum number of activities

B. A problem that minimizes the cost of goods

C. An optimization problem requiring visits to all cities

D. A search algorithm for finding the best path

The Traveling Salesperson Problem (TSP) is fundamentally an optimization problem that requires a salesperson to determine the most efficient route that visits a set of cities and returns to the original city, with the aim of minimizing the total distance or cost associated with the trip. This encapsulates the essence of TSP: it is not merely about visiting all cities, but also about optimizing the path taken while ensuring that each city is visited exactly once before returning to the starting point. This problem is widely studied in various fields, including operations research, computer science, and logistics, because of its practical implications in route planning and resource allocation. The difficulty of TSP arises from its combinatorial nature; as the number of cities increases, the number of possible routes grows factorially, making it challenging to solve efficiently. Thus, the correct answer focuses on the comprehensive visit to all cities while also considering the optimization aspect, which is central to the definition and goal of the TSP.

SAMPLE

10. Which of the following is NOT a characteristic of Merge Sort?

A. It sorts elements by merging sorted halves

B. It requires space for temporary arrays during processing

C. It operates efficiently on large, unsorted lists

D. It performs better than Quick Sort in all cases

Merge Sort is a well-known sorting algorithm characterized by its divide-and-conquer approach. It sorts elements by recursively dividing the list into smaller sublists until each sublist contains a single element and then merging these sublists to produce a sorted list. This characteristic is fundamental to Merge Sort's operation and underscores the correctness of options related to its sorting methodology. The merging process requires additional memory, as temporary arrays are created to hold the elements being merged. This aligns with the characteristic of Merge Sort needing space for these temporary arrays during its processing, which indicates that understanding the algorithm's space requirements is crucial. Merge Sort is particularly efficient for large, unsorted lists due to its predictable time complexity of O(n log n), regardless of the input data. This makes it suitable for handling large volumes of data where consistent performance is essential. However, it is important to note that Merge Sort does not always perform better than Quick Sort in all scenarios. Quick Sort is generally faster on average because it has a lower constant factor in its time complexity. Although Quick Sort can degrade to $O(n^2)$ in its worst-case scenario, this condition is less common compared to Merge Sort's O(n log n) performance. Therefore, stating that Merge Sort performs better than Quick Sort in

SAMPLE

Next Steps

Congratulations on reaching the final section of this guide. You've taken a meaningful step toward passing your certification exam and advancing your career.

As you continue preparing, remember that consistent practice, review, and self-reflection are key to success. Make time to revisit difficult topics, simulate exam conditions, and track your progress along the way.

If you need help, have suggestions, or want to share feedback, we'd love to hear from you. Reach out to our team at hello@examzify.com.

Or visit your dedicated course page for more study tools and resources:

https://wgu-d334algorithms.examzify.com

We wish you the very best on your exam journey. You've got this!

SAMPLE