Intractability II (P, NP, and NP-complete)

Quiz

Questions

Introduction

Computing complexity. Discover P, where problems find quick solutions, and NP, where verification is speedy. Uncover the enigma of NP-Complete, problems as tough as the hardest in NP. Journey through reductions, revealing the intricate web of problem relationships. Explore the unsolved P vs. NP puzzle, challenging the limits of what we can efficiently compute. Welcome to the world of algorithmic intractability.

Question #1

What is the main distinction between the classes P and NP in computational complexity theory?

a) P is for problems that can be solved by algorithms in polynomial time; NP is for problems that can be solved in exponential time.

b) P is for problems that can be solved by algorithms in exponential time; NP is for problems that can be solved in polynomial time.

c) P is for problems that can be solved by algorithms in logarithmic time; NP is for problems that can be solved in linear time.

d) P is for problems that can be solved by algorithms in linear time; NP is for problems for which solutions can be verified in polynomial time.

Answer : d) P is for problems that can be solved by algorithms in linear time; NP is for problems for which solutions can be verified in polynomial time.

Question #2

Which class contains problems that are at least as hard as the hardest problems in NP and has the property that if any problem in this class can be solved in polynomial time, then all problems in NP can be solved in polynomial time?

a) P
b) NP
c) NP-Complete
d) PSPACE

Answer 2: c) NP-Complete

Question #3

State one problem that is widely recognized as an NP-Complete problem and briefly explain its significance.

a) Sudoku puzzle; used for cryptographic applications.
b) Traveling Salesman Problem; helps in optimizing routes and logistics.
c) Sorting a list of numbers; essential for database management.
d) Finding the shortest path in a tree; used in computer graphics rendering.

Answer 3: b) Traveling Salesman Problem; helps in optimizing routes and logistics.

Question #4

What is the key concept behind reductions in the context of NP-Complete problems?

a) Reducing the runtime of an algorithm.
b) Transforming one problem into another.
c) Reducing the complexity of an algorithm.
d) Increasing the efficiency of an algorithm.

Answer 4: b) Transforming one problem into another.

Question #5

What is the current status of the P vs. NP problem, and why is it important in computer science?

a) The problem has been solved, showing that P = NP, and this has revolutionized cryptography.
b) The problem has been solved, showing that P ≠ NP, and this has implications for optimization algorithms.
c) The problem is open, and its solution could have profound implications for the efficiency of algorithms and what can be computed efficiently.
d) The problem is open, but it is considered unimportant in computer science due to its theoretical nature

Answer 5: c) The problem is open, and its solution could have profound implications for the efficiency of algorithms and what can be computed efficiently.