- 1. What are the sources of optimization for basic blocks in code?
- a) Loop unrolling
- b) Instruction scheduling
- c) Register allocation
- d) All of the above
- Answer: d) All of the above

Explanation: Basic block optimizations involve various techniques like loop unrolling to reduce loop overhead, instruction scheduling to improve instruction throughput, and register allocation to minimize memory accesses.

2. Which optimization technique is primarily used for optimizing loops in flow graphs?

- a) Dead code elimination
- b) Loop unrolling
- c) Common subexpression elimination
- d) Data flow analysis

Answer: b) Loop unrolling

Explanation: Loop unrolling is a technique used to optimize loops by reducing loop control overhead and exposing more instruction-level parallelism.

3. What does dead code elimination aim to accomplish in code optimization?

- a) Removing unreachable code
- b) Optimizing loops
- c) Minimizing register pressure
- d) Improving memory access patterns

Answer: a) Removing unreachable code

Explanation: Dead code elimination identifies and removes code that cannot be executed, thus reducing code size and potentially improving runtime performance.

4. Which of the following is a key aspect of loop optimization?

- a) Reducing loop iterations
- b) Introducing more loop nests
- c) Increasing loop overhead
- d) Improving data locality
- Answer: d) Improving data locality

Explanation: Loop optimization aims to improve performance by enhancing data locality, reducing loop overhead, and maximizing instruction-level parallelism.

5. What is the primary goal of global data flow analysis in code optimization?

- a) Identifying unused variables
- b) Minimizing control flow
- c) Analyzing variable usage across the entire program
- d) Improving cache utilization

Answer: c) Analyzing variable usage across the entire program

Explanation: Global data flow analysis examines how variables are used across the entire program to optimize register allocation, identify dead code, and improve overall performance.

6. Which type of transformation is commonly used to improve code performance by rearranging instructions without changing program semantics?

- a) Loop unrolling
- b) Instruction scheduling
- c) Dead code elimination
- d) Common subexpression elimination
- Answer: b) Instruction scheduling

Explanation: Instruction scheduling involves rearranging instructions to minimize pipeline stalls and improve instruction throughput without altering the program's functionality.

7. What is the purpose of symbolic debugging of optimized code?

- a) Identifying performance bottlenecks
- b) Analyzing memory usage
- c) Improving code readability
- d) Debugging without losing optimization benefits
- Answer: d) Debugging without losing optimization benefits

Explanation: Symbolic debugging allows developers to debug optimized code while preserving the optimizations applied, enabling efficient debugging without sacrificing performance.

8. Which optimization technique focuses on identifying and eliminating redundant computations?

- a) Loop unrolling
- b) Common subexpression elimination
- c) Register allocation
- d) Dead code elimination
- Answer: b) Common subexpression elimination

Explanation: Common subexpression elimination removes redundant computations by identifying expressions that have already been computed and stored, thereby reducing redundant computations.

9. What is the primary purpose of loop unrolling?

- a) Decreasing code size
- b) Reducing loop overhead
- c) Introducing more loops
- d) Minimizing data dependencies
- Answer: b) Reducing loop overhead

Explanation: Loop unrolling aims to reduce loop overhead by executing multiple loop iterations in a single iteration, thereby improving performance.

10. Which optimization technique focuses on utilizing CPU registers efficiently?

- a) Loop unrolling
- b) Instruction scheduling
- c) Register allocation
- d) Dead code elimination
- Answer: c) Register allocation

Explanation: Register allocation assigns variables to CPU registers to minimize memory accesses, reduce register pressure, and improve overall performance.

- 11. What does loop fusion involve in optimizing loops?
- a) Combining multiple loops into a single loop
- b) Breaking a loop into smaller loops

- c) Reversing loop iterations
- d) Introducing more loop nests
- Answer: a) Combining multiple loops into a single loop

Explanation: Loop fusion combines multiple loops into a single loop to reduce loop overhead and improve data locality.

12. Which optimization technique focuses on rearranging code to exploit instruction-level parallelism?

- a) Instruction scheduling
- b) Loop unrolling
- c) Dead code elimination
- d) Common subexpression elimination
- Answer: a) Instruction scheduling

Explanation: Instruction scheduling rearranges code to exploit instruction-level parallelism and minimize pipeline stalls, thus improving performance.

13. What is the primary goal of loop-invariant code motion?

- a) Reducing loop iterations
- b) Moving computations outside the loop
- c) Increasing loop overhead
- d) Introducing more loop nests

Answer: b) Moving computations outside the loop

Explanation: Loop-invariant code motion moves computations that do not change during loop execution outside the loop to reduce redundant computations and improve performance.

14. Which optimization technique aims to eliminate code that does not affect program output?

- a) Loop unrolling
- b) Dead code elimination
- c) Register allocation
- d) Instruction scheduling
- Answer: b) Dead code elimination

Explanation: Dead code elimination identifies and removes code that does not affect program output, reducing code size and potentially improving performance.

15. What is the primary purpose of loop peeling in loop optimization?

- a) Reducing loop overhead
- b) Increasing loop iterations
- c) Introducing more loop nests
- d) Improving data locality
- Answer: a) Reducing loop overhead

Explanation: Loop peeling involves removing the first few iterations of a loop to reduce loop overhead and improve performance.

16. Which optimization technique focuses on minimizing memory access patterns?

- a) Loop unrolling
- b) Register allocation
- c) Dead code elimination
- d) Data prefetching

Answer: d) Data prefetching

Explanation: Data prefetching aims to minimize memory access latency by fetching data into cache before it is actually needed, thereby improving performance.

17. What does loop interchange involve in loop optimization?

- a) Reversing loop iterations
- b) Exchanging the order of nested loops
- c) Combining multiple loops into a single loop
- d) Introducing more loop nests
- Answer: b) Exchanging the order of nested loops

Explanation: Loop interchange exchanges the order of nested loops to improve data locality and reduce loop overhead, thereby enhancing performance.

18. Which optimization technique aims to reduce the number of instructions executed inside loops?

- a) Loop unrolling
- b) Loop fusion
- c) Loop-invariant code motion
- d) Loop interchange
- Answer: c) Loop-invariant code motion

Explanation: Loop-invariant code motion moves computations outside loops to reduce the number of instructions executed inside loops, thereby improving performance.

19. What is the primary purpose of loop vectorization?

a) Minimizing loop overhead

b) Introducing more loop nests

c) Exploiting SIMD instructions for parallel execution

d) Eliminating redundant computations

Answer: c) Exploiting SIMD instructions for parallel execution

Explanation: Loop vectorization transforms scalar operations into SIMD operations to exploit parallelism and improve performance.

20. Which optimization technique focuses on reducing the number of conditional branches in code?

a) Loop unrolling

b) Dead code elimination

- c) Loop fusion
- d) Control flow optimization

Answer: d) Control flow optimization

Explanation: Control flow optimization aims to reduce the number of conditional branches in code to improve branch prediction accuracy and overall performance.

Related posts:

- 1. Introduction to compiling & Lexical Analysis MCQs
- 2. Syntax Analysis & Syntax Directed Translation MCQs
- 3. Type Checking & Run Time Environment MCQs
- 4. Code Generation MCQs
- 5. Introduction to Energy Science MCQ
- 6. Ecosystems MCQ
- 7. Biodiversity and its conservation MCQ

- 8. Environmental Pollution mcq
- 9. Social Issues and the Environment MCQ
- 10. Field work mcq
- 11. Discrete Structure MCQ
- 12. Set Theory, Relation, and Function MCQ
- 13. Propositional Logic and Finite State Machines MCQ
- 14. Graph Theory and Combinatorics MCQ
- 15. Relational algebra, Functions and graph theory MCQ
- 16. Data Structure MCQ
- 17. Stacks MCQ
- 18. TREE MCQ
- 19. Graphs MCQ
- 20. Sorting MCQ
- 21. Digital Systems MCQ
- 22. Combinational Logic MCQ
- 23. Sequential logic MCQ
- 24. Analog/Digital Conversion, Logic Gates, Multivibrators, and IC 555 MCQ
- 25. Introduction to Digital Communication MCQ
- 26. Introduction to Object Oriented Thinking & Object Oriented Programming MCQ
- 27. Encapsulation and Data Abstraction MCQ
- 28. MCQ
- 29. Relationships Inheritance MCQ
- 30. Polymorphism MCQ
- 31. Library Management System MCQ
- 32. Numerical Methods MCQ
- 33. Transform Calculus MCQ
- 34. Concept of Probability MCQ

- 35. Algorithms, Designing MCQ
- 36. Study of Greedy strategy MCQ
- 37. Concept of dynamic programming MCQ
- 38. Algorithmic Problem MCQ
- 39. Trees, Graphs, and NP-Completeness MCQ
- 40. The Software Product and Software Process MCQ
- 41. Software Design MCQ
- 42. Software Analysis and Testing MCQ
- 43. Software Maintenance & Software Project Measurement MCQ
- 44. Computer Architecture, Design, and Memory Technologies MCQ
- 45. Basic Structure of Computer MCQ
- 46. Computer Arithmetic MCQ
- 47. I/O Organization MCQ
- 48. Memory Organization MCQ
- 49. Multiprocessors MCQ
- 50. Introduction to Operating Systems MCQ
- 51. File Systems MCQ
- 52. CPU Scheduling MCQ
- 53. Memory Management MCQ
- 54. Input / Output MCQ
- 55. Operating Systems and Concurrency
- 56. Software Development and Architecture MCQ
- 57. Software architecture models MCQ
- 58. Software architecture implementation technologies MCQ
- 59. Software Architecture analysis and design MCQ
- 60. Software Architecture documentation MCQ
- 61. Introduction to Computational Intelligence MCQ

- 62. Fuzzy Systems MCQ
- 63. Genetic Algorithms MCQ
- 64. Rough Set Theory MCQ
- 65. Introduction to Swarm Intelligence, Swarm Intelligence Techniques MCQ
- 66. Neural Network History and Architectures MCQ
- 67. Autoencoder MCQ
- 68. Deep Learning MCQs
- 69. RL & Bandit Algorithms MCQs
- 70. RL Techniques MCQs
- 71. Review of traditional networks MCQ
- 72. Study of traditional routing and transport MCQ
- 73. Wireless LAN MCQ
- 74. Mobile transport layer MCQ
- 75. Big Data MCQ
- 76. Hadoop and Related Concepts MCQ
- 77. Hive, Pig, and ETL Processing MCQ
- 78. NoSQL MCQs Concepts, Variations, and MongoDB
- 79. Mining social Network Graphs MCQ
- 80. Mathematical Background for Cryptography MCQ
- 81. Cryptography MCQ
- 82. Cryptographic MCQs
- 83. Information Security MCQ
- 84. Cryptography and Information Security Tools MCQ
- 85. Data Warehousing MCQ
- 86. OLAP Systems MCQ
- 87. Introduction to Data& Data Mining MCQ
- 88. Supervised Learning MCQ

- 89. Clustering & Association Rule mining MCQ
- 90. Fundamentals of Agile Process MCQ
- 91. Agile Projects MCQs
- 92. Introduction to Scrum MCQs
- 93. Introduction to Extreme Programming (XP) MCQs
- 94. Agile Software Design and Development MCQs
- 95. Machine Learning Fundamentals MCQs
- 96. Neural Network MCQs
- 97. CNNs MCQ
- 98. Reinforcement Learning and Sequential Models MCQs
- 99. Machine Learning in ImageNet Competition mcq
- 100. Computer Network MCQ
- 101. Data Link Layer MCQ
- 102. MAC Sub layer MCQ
- 103. Network Layer MCQ
- 104. Transport Layer MCQ
- 105. Raster Scan Displays MCQs
- 106. 3-D Transformations MCQs
- 107. Visualization MCQ
- 108. Multimedia MCQs
- 109. INTRODUCTION Knowledge Management MCQs
- 110. Organization and Knowledge Management MCQs
- 111. Telecommunications and Networks in Knowledge Management MCQs
- 112. Components of a Knowledge Strategy MCQs
- 113. Advanced topics and case studies in knowledge management MCQs
- 114. Conventional Software Management MCQs
- 115. Software Management Process MCQs

- 116. Software Management Disciplines MCQs
- 117. Rural Management MCQs
- 118. Human Resource Management for rural India MCQs
- 119. Management of Rural Financing MCQs
- 120. Research Methodology MCQs
- 121. Research Methodology MCQs
- 122. IoT MCQs
- 123. Sensors and Actuators MCQs
- 124. IoT MCQs: Basics, Components, Protocols, and Applications
- 125. MCQs on IoT Protocols
- 126. IoT MCQs
- 127. INTRODUCTION Block Chain Technologies MCQs
- 128. Understanding Block chain with Crypto currency MCQs
- 129. Understanding Block chain for Enterprises MCQs
- 130. Enterprise application of Block chain MCQs
- 131. Block chain application development MCQs
- 132. MCQs on Service Oriented Architecture, Web Services, and Cloud Computing
- 133. Utility Computing, Elastic Computing, Ajax MCQs
- 134. Data in the cloud MCQs
- 135. Cloud Security MCQs
- 136. Issues in cloud computinG MCQs
- 137. Introduction to modern processors MCQs
- 138. Data access optimizations MCQs
- 139. Parallel Computing MCQs
- 140. Efficient Open MP Programming MCQs
- 141. Distributed Memory parallel programming with MPI MCQs
- 142. Review of Object Oriented Concepts and Principles MCQs.

- 143. Introduction to RUP MCQs.
- 144. UML and OO Analysis MCQs
- 145. Object Oriented Design MCQs
- 146. Object Oriented Testing MCQs
- 147. CVIP Basics MCQs
- 148. Image Representation and Description MCQs
- 149. Region Analysis MCQs
- 150. Facet Model Recognition MCQs
- 151. Knowledge Based Vision MCQs
- 152. Game Design and Semiotics MCQs
- 153. Systems and Interactivity Understanding Choices and Dynamics MCQs
- 154. Game Rules Overview Concepts and Case Studies MCQs
- 155. IoT Essentials MCQs
- 156. Sensor and Actuator MCQs
- 157. IoT Networking & Technologies MCQs
- 158. MQTT, CoAP, XMPP, AMQP MCQs
- 159. IoT MCQs: Platforms, Security, and Case Studies
- 160. MCQs on Innovation and Entrepreneurship
- 161. Innovation Management MCQs
- 162. Stage Gate Method & Open Innovation MCQs
- 163. Innovation in Business: MCQs
- 164. Automata Theory MCQs
- 165. Finite Automata MCQs
- 166. Grammars MCQs
- 167. Push down Automata MCQs
- 168. Turing Machine MCQs
- 169. Database Management System (DBMS) MCQs

- 170. Relational Data models MCQs
- 171. Data Base Design MCQs
- 172. Transaction Processing Concepts MCQs
- 173. Control Techniques MCQs
- 174. DBMS Concepts & SQL Essentials MCQs
- 175. DESCRIPTIVE STATISTICS MCQs
- 176. INTRODUCTION TO BIG DATA MCQ
- 177. BIG DATA TECHNOLOGIES MCQs
- 178. PROCESSING BIG DATA MCQs
- 179. HADOOP MAPREDUCE MCQs
- 180. BIG DATA TOOLS AND TECHNIQUES MCQs
- 181. Pattern Recognition MCQs
- 182. Classification Algorithms MCQs
- 183. Pattern Recognition and Clustering MCQs
- 184. Feature Extraction & Selection Concepts and Algorithms MCQs
- 185. Pattern Recognition MCQs
- 186. Understanding Cybercrime Types and Challenges MCQs
- 187. Cybercrime MCQs
- 188. Cyber Crime and Criminal justice MCQs
- 189. Electronic Evidence MCQs
- 190. Array MCQS
- 191. Unix/Linux MCQs
- 192. Biodiversity and its conservation MCQs
- 193. Frequency domain representation of signal mcqs
- 194. State Space & Control Systems MCQs
- 195. The z-Transformmcqs
- 196. Propagation of radio waves mcqs

- 197. Satellite Systems and Orbital Mechanics MCQs
- 198. Embedded System Architecture mcqs
- 199. Rectifiers and Thyristors MCQs
- 200. CMOS Processing Technology MCQs