VALLIAMMAI ENGINEERING COLLEGE
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
ADVANCED DATASTRUCTURE AND ALGORITHM
QUESTION BANK

UNIT I ITERATIVE AND RECURSIVE ALGORITHMS
PART-A

1. Define iterative algorithm
2. What is sequence of action and sequence of assertion
3. Define precondition
4. Define Post condition
5. Define loop invariant
6. List the types of iterative algorithm.
7. List the technical error in iterative algorithm.
8. What is proof of correctness of each step in iterative algorithm?
9. Give an example with more of input and more of output.
10. Define Exit condition
11. Define recursive algorithm
12. Define stack frame
13. Define tree of stack frame
14. Write algorithm for Tower of Honai.
15. Write algorithm for Acremann
16. What is forward and backward recursive?
17. Write algorithm for power of integer.
18. What is sub instance?
19. What is sub solution?
20. Write a algorithm for three tree traversal method.

PART-B

1. Explain briefly about steps to develop iterative algorithm.
2. Explain briefly about more of input with example
3 Explain briefly about more of output with example
4. Explain briefly about steps to develop recursive algorithm
5. Explain briefly with Towers of Hanoi
6. Explain the Check list for recursive Algorithm
7. Explain briefly with power of integer
8. Explain briefly with sorting algorithm
9. Explain briefly with searching algorithm
10. Explain briefly about Heap sort
11. Explain briefly about Tree Traversal

UNIT-II OPTIMIZATION ALGORITHMS

PART-A

1. what is order of handling nodes
2. what is optimization algorithms.?
3. How to prove path is shortest.
4. List the classification of edges.
5. What is pruning path?
6. What is bipartite?
7. What is forward edges and cross edges?

8. Definition of Total order.
9. Definition of Partial order.
10. What is min cut specification?
11. What is network flow?
12. What is Augmentation path.?
13. What is recursive backtracking?
14. What is little bird?
15. What is pruning branch?
16. What is satisfiability?
17. List the steps to fixing the algorithm.
18. What is local maximum and global maximum?
19. What is max flow?
20. What is linear programming?

PART-B

1. Explain briefly about BFS,
2. Explain briefly about DFS,
3. Explain briefly about Dijkstras shortest weighted path
4. Explain briefly about generic search algorithm
5. Explain briefly about recursive DFS.
6. Explain briefly about network flows and linear programming
7. Explain Hill climbing algorithm with small local maximum,
8. Explain about primal dual hill climbing
9. Explain about steepest-ascent hill climbing
10. Explain about pruning branches satisfiability

UNIT-III DYNAMIC PROGRAMMING ALGORITHMS

Part-A

1. what is dynamic programming
2. What is level graph?
3. What is question of little bird?
4. What is print neatly problem?
5. What is reduction problem?
6. What is the classification made based on reductions?
7. What is circuit satisfiability problem?
8. What is bipartite matching?
9. What is memorization?
10. List the set of sub instance
11. List the three example for NP Complete problem
12. What is NP complete problem?
13. What is randomized algorithm?
14. What is expander graph?
15. What is graph colouring?

Part-B

1. Explain briefly about Dynamic programming
2. Explain briefly about little bird,sub instances and sub-solution,

3 Explain briefly about np-completeness and proving np-complete

4 Explain about 3 coloring

5. Explain about bipartite
6. Explain about randomized algorithm
7. Explain on how to decrease time and space in dynamic programming problems.
8. Discuss the steps needed to prove NP completeness and apply it to the 3 Coloring problem.

UNIT-IV SHARED OBJECTS AND CONCURRENT OBJECTS

Part-A

1. What is shared object?
2. What is synchronization
3. what is mutual exclusion
4. what is dead lock freedom
5. what is starvation freedom
6. List the communication occur in concurrent system.
7. What is producer and consumer problem?
8. What id reader and writer problem.
9. What is Amdal law
10. What is parallel programming?
11. What is start of danger zone and end of danger zone?
12. List the two thread solution.
13. What is Peterson lock.?
14. What is filter lock?
15. Define fairness.
16. Define Lock object state.
17. Define covering state
18. What is bounded timestamps?

Part-B

1. Explain briefly about producer consumer.
2. Explain briefly about reader and writers
3. Explain briefly about mutual exclusion.
4. Explain briefly about critical section,
5. Explain briefly lamports bakery,
6. Explain concurrent objects, concurrency and correctness,
7. Explain briefly about java memory model
8. Write an algorithm for filter lock mutual exclusion protocol and show how it achieves mutual exclusion property.
9. What is Linearizability? Explain with example.
10. is Java Supports concurrency, why or why?

UNIT-V CONCURRENT DATA STRUCTURES

Part-A

1. What is coarse gained synchronization?
2. What is fine grained synchronization?
3. What is optimization synchronization?
4. What is list based set?
5. What is locking and unlocking?
6. What is bounded partial queue?
7. What is unbounded total queue?
8. What is unbounded lock free queue?
9. What is naive synchronization queue?
10. What is elimination back of stack?
11. What is elimination array?
12. What is lazy synchronization?
13. What is non-Blocking synchronization?
14. What is dual datastructure?
15. What is elimination back of stack?
16. What are different varieties of pool?

Part-B

1. Explain briefly about Types of synchronization(5 types is there)
2. Explain briefly about Unbounded lock free queue and stack
3. Explain about Concurrent stack

4 Explain about concurrent queue.

5. What is ABA problem? How it related with memory reclamation, show the steps in the process of memory reclamation using diagram and algorithm?
6. Perform Push and pop operation in an unbounded lock free stack with the help of code.
7. would the lazy algorithm still work if we marked a node as removed simply by setting its next field to null? Why or why not?what about the lock free algorithm?
8. The add() method of the lock free algorithm never finds a marked node with tha same key.can the algorithm be modified so that it will simply insert its new added object into the existing marked node with same key if such a node exits in the list, thus saving the need to insert a new node?

--------------------------------------- Hard Work never fails ---