QUIZ 3: Operations Research - Expert & Application Level

Network Algorithms and Graph Theory

Course: USEEN3 - Operations Research
Master: Networks-IoT
Academic Year: 2025-2026
Total Questions: 70

Instructions

This quiz focuses on:

- Edge cases and boundary conditions
- Theoretical proofs and correctness arguments
- Real-world applications and trade-offs
- Advanced algorithm variations

Assumptions: n = vertices, m = edges (or arcs), w(e) = edge weight

Section 1: Correctness & Proof-Based Questions (10 questions)

1. Why is the greedy approach in Dijkstra's algorithm correct?

A) It always picks the smallest edge
B) Once dist[v] is finalized, no shorter path exists
C) All vertices are processed in order
D) It maintains a priority queue

Answer: B

2. Prove that Bellman-Ford correctly detects negative cycles by checking if after n-1 iterations, an edge can still be relaxed.

Which statement is the theoretical basis?

A) Any path with n vertices has n-1 edges
B) Negative cycles allow infinite relaxation
C) Simple paths in a negative cycle have n vertices
D) All of the above

Answer: D

3. The Cut Property for MST states:

"For any cut of graph, the minimum weight edge crossing the cut is in **some** MST."

Why "some" and not "the"?

A) There can be multiple MSTs
B) Different cuts may prefer different edges
C) Edge weights might be equal
D) All of the above

Answer: D

4. Why must Prim's algorithm start with any single vertex?

A) To ensure connectedness of the growing tree
B) To avoid isolated components
C) Because MST includes all vertices
D) All of the above

Answer: D

5. In Kruskal's algorithm, why is it safe to add an edge if it doesn't form a cycle?

A) The union-find data structure guarantees this
B) By the cycle property of MST
C) Because we process edges in order
D) A and B

Answer: D

6. For topological sort to exist in a DAG:

Which statement is FALSE?

A) The graph must be acyclic
B) Every DAG has at least one source vertex
C) Multiple valid topological orderings may exist
D) A unique ordering always exists

Answer: D

7. Why does BFS find the shortest path in unweighted graphs but DFS doesn't?

A) BFS uses a queue, DFS uses a stack
B) BFS explores level-by-level guaranteeing shortest path found first
C) DFS may traverse longer paths first
D) All of the above

Answer: D

8. The correctness of Bellman-Ford relies on the principle that:

A) A shortest path has at most n-1 edges in a simple path
B) After k iterations, shortest paths using $\leq k$ edges are found
C) Relaxation monotonically improves distances
D) All of the above

Answer: D

9. Why is the adjacency list representation better for sparse graphs?

A) Space savings: $O(n+m)$ vs $O(n^2)$
B) Traversal efficiency: iterate only over actual edges
C) Both A and B
D) Adjacency matrix is actually better

Answer: C

10. Prove that in an undirected tree with n vertices: the sum of all degrees = 2(n-1)

Using the handshaking lemma: $\sum \text{degree}(v) = 2m = 2(n-1)$

Why is m = n-1 for a tree?

A) Definition of a tree
B) Theorem: connected acyclic graph
C) Minimum edges for connectivity
D) All of the above

Answer: D

Section 2: Edge Cases & Boundary Conditions (12 questions)

11. Consider a graph with a single vertex and no edges. Is it:

A) A tree
B) Connected
C) An MST of itself
D) All of the above

Answer: D

12. In Dijkstra's algorithm with a single source, if the graph is disconnected:

A) Unreachable vertices have dist = ∞
B) Algorithm terminates normally
C) Those vertices are never added to the finished set
D) All of the above

Answer: D

13. When there are multiple edges with the same minimum weight in Kruskal:

A) The order doesn't matter for MST weight
B) Different MSTs may result
C) The choice affects the final MST weight
D) A and B

Answer: D

14. If all edge weights in a graph are equal:

A) All spanning trees have the same weight
B) Any spanning tree is an MST
C) The MST is unique
D) A and B only

Answer: D

15. In BFS/DFS on a graph with multiple connected components:

A) We must restart from each unvisited vertex
B) We visit all components eventually
C) The starting vertex matters for which component is first
D) All of the above

Answer: D

16. For a complete graph K_n with n=2:

A) It has 1 edge
B) It is a tree
C) It is its own MST
D) All of the above

Answer: D

17. In Bellman-Ford, if a negative cycle exists but is unreachable from source s:

A) Algorithm still detects it
B) Algorithm doesn't detect it
C) Distances to unreachable vertices remain ∞
D) B and C

Answer: D

18. A DAG with n vertices must have:

A) At least one sink vertex (outdegree 0)
B) At least one source vertex (indegree 0)
C) Both A and B (unless n=1)
D) Neither necessarily

Answer: C

19. In a graph where all edges have weight 0:

A) BFS finds shortest paths in O(n+m)
B) Dijkstra still works
C) A and B
D) Bellman-Ford is required

Answer: C

20. When a vertex has degree 0 in a connected graph:

A) It's impossible (contradiction)
B) The graph is not connected
C) It must be isolated
D) All of the above

Answer: D

21. Self-loops (edge from vertex to itself):

In adjacency matrix, they contribute:

A) 1 to the matrix
B) 2 to the degree
C) 0 to the degree
D) Depends on whether it's directed

Answer: D

22. A directed graph with no arcs has how many strongly connected components?

A) 0
B) 1
C) n (one per vertex)
D) m

Answer: C

Section 3: Algorithm Variations & Optimizations (12 questions)

23. The A* algorithm is a variant of Dijkstra that uses:

A) A heuristic function
B) Estimated cost to goal
C) Can find shortest paths faster with good heuristics
D) All of the above

Answer: D

24. Bidirectional search from source s and target t:

A) Searches from both ends toward middle
B) Potentially reduces search space
C) Can find shortest path when they meet
D) All of the above

Answer: D

25. Johnson's algorithm for all-pairs shortest paths:

A) Uses Bellman-Ford once then Dijkstra n times
B) Handles negative weights
C) Achieves $O(n^2 \log n + nm)$ with good implementation
D) All of the above

Answer: D

26. In hierarchical/level-based routing (used by ISPs):

A) Different areas have different routing details
B) Reduces routing table size
C) Trades optimality for scalability
D) All of the above

Answer: D

27. Lazy Prim's algorithm differs from eager Prim by:

A) Not using a priority queue
B) Allowing multiple copies of vertices in the queue
C) Valid updates are ignored until extraction
D) All of the above

Answer: D

28. In distributed Bellman-Ford (used in RIP):

A) Each router maintains distance vector
B) Updates sent to neighbors periodically
C) May take long time to converge with changes
D) All of the above

Answer: D

29. Contraction hierarchies for road networks:

A) Pre-process graph to speed up queries
B) Create shortcuts between important vertices
C) Can answer distance queries very quickly
D) All of the above

Answer: D

30. When using a Fibonacci heap in Dijkstra:

A) Insert: O(1) amortized
B) Extract-min: O(log n) amortized
C) Decrease-key: O(1) amortized
D) All of the above

Answer: D

31. Dial's algorithm for shortest paths:

A) Works on graphs with small integer weights
B) Achieves O(m + nW) where W is max weight
C) Uses an array of buckets instead of priority queue
D) All of the above

Answer: D

32. In the "hub labels" technique:

A) Pre-compute shortest paths to hub vertices
B) Query time becomes O(number of hubs)
C) Very fast in practice with preprocessing
D) All of the above

Answer: D

33. Parallel algorithms for MST:

A) Borůvka's algorithm naturally parallelizes
B) Kruskal requires sorting which parallels well
C) Different components can be processed in parallel
D) All of the above

Answer: D

34. Dynamic shortest path algorithms for graphs with changing weights:

A) May use the previous solution as a starting point
B) Don't need to recompute from scratch
C) Can use decremental or incremental updates
D) All of the above

Answer: D

Section 4: Real-World Applications & Trade-offs (12 questions)

35. In GPS navigation, which algorithm is most appropriate?

A) Dijkstra for exact shortest paths
B) A* with heuristic to actual destination
C) Bidirectional A* for faster computation
D) B or C, depending on implementation

Answer: D

36. In social network analysis, finding shortest paths between users requires:

A) Handling potentially disconnected components
B) Very large graphs (billions of vertices)
C) Fast queries but not necessarily optimal
D) All of the above

Answer: D

37. In a datacenter network, finding minimum spanning tree for broadcast:

A) Reduces bandwidth usage
B) Creates a tree structure for flooding
C) Important for protocol efficiency
D) All of the above

Answer: D

38. RIP's limitation to 15 hops came from:

A) Hardware constraints at protocol design
B) To prevent count-to-infinity problems
C) Trade-off between scalability and routing table size
D) A and C

Answer: D

39. OSPF's use of Areas is to:

A) Reduce the size of routing tables
B) Limit LSA flooding scope
C) Hierarchical organization of network
D) All of the above

Answer: D

40. In BGP routing (between ASes):

A) Path-vector protocol carries entire path
B) Detects routing loops via path inspection

C) Allows policy-based routing decisions
D) All of the above

Answer: D

41. Load balancing using multiple shortest paths:

A) Requires finding all shortest paths
B) May distribute traffic across them
C) Improves network utilization
D) All of the above

Answer: D

42. In underwater cable networks:

A) MST minimizes total cable length
B) Connectivity costs are very high
C) Redundancy is important despite MST
D) All of the above

Answer: D

43. For real-time traffic routing:

A) Algorithms must be extremely fast
B) Exact optimality may be sacrificed
C) Caching/prediction helps
D) All of the above

Answer: D

44. In software-defined networks (SDN):

A) Centralized controller computes paths
B) Can use optimal algorithms offline
C) Switches follow computed forwarding rules
D) All of the above

Answer: D

45. Quantum networks may require:

A) New shortest path algorithms
B) Exploiting quantum properties
C) Different optimization objectives
D) All possible

Answer: D

46. In resilient networking:

A) Need to handle link/node failures
B) May want k shortest paths not just shortest
C) Requires updated algorithms
D) All of the above

Answer: D

Section 5: Mathematical & Theoretical Questions (12 questions)

47. The spanning tree of a connected graph G has exactly:

A) n-1 edges (always)
B) n edges (sometimes)
C) n+1 edges (rarely)
D) Depends on graph

Answer: A

48. In a forest with k trees containing n vertices total:

A) Total edges = n-k
B) Total edges = n+k
C) Total edges = n/k
D) Cannot determine

Answer: A

49. A bipartite graph:

A) Can be colored with 2 colors
B) Has no odd-length cycles
C) Contains no triangles
D) All of the above

Answer: D

50. The chromatic number of a complete graph K_n:

A) 1
B) n
C) n-1
D) Cannot determine

Answer: B

51. In a tournament graph (directed complete graph):

A) Every pair of vertices has exactly one directed edge
B) Every vertex has indegree + outdegree = n-1

C) There always exists a Hamiltonian path
D) All of the above

Answer: D

52. Two vertices u and v are in the same SCC if and only if:

A) There is a path from u to v
B) There is a path from v to u
C) Both A and B
D) One of A or B

Answer: C

53. The condensation graph (DAG of SCCs):

A) Is always a DAG
B) Has no cycles by definition
C) Has fewer vertices than original
D) All of the above

Answer: D

54. For a planar graph with n vertices and m edges:

A) $m \leq 3n - 6$
B) Has at most O(n) edges
C) Can be drawn without edge crossings
D) All of the above

Answer: D

55. The property "no odd-length cycles" characterizes:

A) Bipartite graphs
B) Trees
C) Forests
D) A only

Answer: A

56. For a flow network, the maximum flow equals:

A) The minimum cut (Max-Flow Min-Cut theorem)
B) Sum of all edge capacities
C) Product of vertex capacities
D) A only

Answer: A

57. The independence number $\alpha(G)$ of a graph G:

A) Size of maximum independent set
B) No two vertices are adjacent in the set
C) For a tree, can be ≥ n/2
D) All of the above

Answer: D

58. Dilworth's theorem states:

A) Every partially ordered set can be partitioned
B) Number of chains equals width of poset
C) Related to longest antichain
D) All of the above

Answer: D

Section 6: Comparative Analysis & Trade-offs (12 questions)

59. Comparing BFS vs DFS:

Aspect	BFS	DFS
Memory for dense graph	O(n)	O(n)
Shortest path (unweighted)	Yes	No
Time complexity	O(n+m)	O(n+m)
Optimal for trees	Both	Both

Which is true?

A) All cells above are correct
B) BFS uses O(n) queue
C) Both find shortest paths on trees
D) All true

Answer: A

60. Dijkstra vs Bellman-Ford:

Choose TRUE statements: *(Multiple answers)*

A) Dijkstra faster for non-negative weights
B) Bellman-Ford handles negative weights
C) Both O(nm) when implemented naively
D) Dijkstra fails on negative cycles

Answer: A, B, D

61. Prim vs Kruskal for MST:

A) Prim grows one tree; Kruskal grows multiple
B) Kruskal requires sorting; Prim requires heap
C) Both are greedy and correct
D) All of the above

Answer: D

62. For sparse graphs ($m = O(n)$):

A) Dijkstra with heap: $O(n \log n)$
B) Bellman-Ford: $O(n^2)$
C) Dijkstra is much better
D) All true

Answer: D

63. For very dense graphs ($m = \Theta(n^2)$):

A) Dijkstra with heap still $O(n^2 \log n)$
B) Simple $O(n^2)$ implementation of Dijkstra is better
C) Floyd-Warshall $O(n^3)$ may be comparable
D) All of the above

Answer: D

64. Link-state (OSPF) vs Distance-vector (RIP):

A) Link-state sends complete topology
B) Distance-vector sends routing tables
C) Link-state converges faster
D) All of the above

Answer: D

65. Static routing vs Dynamic routing:

A) Static: manually configured routes
B) Dynamic: automatically computed
C) Dynamic adapts to failures and congestion
D) All of the above

Answer: D

66. Routing table size with OSPF Areas vs RIP:

A) Areas reduce routing table size
B) RIP limited by 15-hop diameter
C) OSPF hierarchical structure helps
D) All of the above

Answer: D

67. Time vs Space trade-offs:

A) Adjacency matrix: $O(n^2)$ space, $O(1)$ lookup
B) Adjacency list: $O(n+m)$ space, O(degree) lookup
C) Trade space for speed with preprocessing
D) All of the above

Answer: D

68. Approximation algorithms:

A) Provide good solutions when exact is too slow
B) Greedy often provides 2-approximation for MST
C) Cannot be exact but can be verified
D) All of the above

Answer: D

69. Heuristics and metaheuristics:

A) Used when problem is NP-hard
B) No guarantee of optimality
C) Include genetic algorithms, simulated annealing
D) All of the above

Answer: D

70. Which design choice affects algorithm efficiency most?

A) Choice of data structure (array vs linked list)
B) Choice of algorithm (Dijkstra vs Bellman-Ford)
C) Implementation details (caching, parallelization)
D) All significantly impact performance

Answer: D

Advanced Problem Scenarios (10 extra questions)

Q1. Design an algorithm to find the k shortest paths between two vertices.

Answer Outline:

- Modify Dijkstra to not stop at first shortest
- Or use Yen's algorithm
- Maintain k shortest paths to each vertex
- Complexity: $O(k \cdot m \log n)$

Q2. How would you compute MST if edge weights are modified dynamically?

Answer Outline:

- Incremental MST: use swap property
- Decremental MST: more complex
- Event-based recomputation
- Or rebuild from scratch (trade-off analysis needed)

Q3. Explain how to detect all bridges in an undirected graph.

Answer Outline:

- Use DFS with discovery times
- Bridge: edge where no back edge goes through
- Articulation point: vertex whose removal disconnects graph
- Time: O(n+m) with single DFS

Q4. How to find maximum flow in a network?

Answer Outline:

- Ford-Fulkerson: O(E·max_flow)
- Edmonds-Karp: $O(VE^2)$
- Dinic's: $O(V^2E)$
- Uses augmenting paths until none remain

Q5. Design a distributed algorithm for finding connected components.

Answer Outline:

- Label propagation: each vertex broadcasts its ID
- Sync rounds until stable
- Similar to minimum spanning tree protocols
- Handles asynchrony and failures

Q6. How does Prim's algorithm work on disconnected graphs?

Answer Outline:

- Start from one component
- Stops at that component's boundary
- Cannot extend to other components
- Run multiple times for each component

Q7. Compare exact vs approximate solutions for traveling salesman problem (TSP).

Answer Outline:

- TSP: NP-hard, no polynomial exact solution
- 2-approximation: MST + shortcutting
- Christofides: 1.5-approximation
- For practical: use heuristics, local search

Q8. How to handle dynamic graphs where vertices/edges are added/removed?

Answer Outline:

- Maintain data structures incrementally
- Recalculate affected portions only
- Cache previous results
- Trade-off between accuracy and update cost

Q9. Explain why some routing protocols diverge or loop initially.

Answer Outline:

- Due to incomplete information initially
- Count-to-infinity problem in RIP
- Trigger updates and hold-down timers
- Link-state avoids this with complete view

Q10. Design redundancy strategy for critical network links using MST concepts.

Answer Outline:

- Build MST for minimum cost connectivity
- Add critical edges: those whose removal disconnects
- Alternative: k-edge-connected subgraph
- Balance cost vs resilience

End of Quiz 3

Key Topics Covered in Quiz 3:

- Correctness proofs and theoretical foundations
- Boundary conditions and edge cases
- Advanced algorithm variations and optimizations
- Real-world applications and practical considerations
- Mathematical and graph-theoretic concepts
- Comparative analysis and trade-offs
- Complex problem scenarios and design questions

Difficulty Progression:

- Questions 1-10: Understanding correctness
- Questions 11-22: Edge cases
- Questions 23-34: Optimizations and variants
- Questions 35-46: Applications
- Questions 47-58: Theory and mathematics
- Questions 59-70: Comparative and trade-offs