

Uber Placement Papers 2026

27 MCQs with answer key · Free download

[Exam pattern](#)

Programming Logic MCQs for DSA themes (graphs, DP, arrays) plus Technical and light Quant/Reasoning. Live OA is multi-problem coding. Practice set — not an official Uber paper.

Questions

[Programming Logic](#)

Q1. Shortest path in city graph with non-negative ETAs:

- A) Dijkstra
- B) DFS always
- C) Bubble
- D) Counting sort

Q2. BFS levels from source give:

- A) Hop distance unweighted
- B) MST weight
- C) Edit distance
- D) Hash load

Q3. Ride intervals; max overlapping rides needs:

- A) Sweep line / sort ends
- B) DFS
- C) KMP
- D) Trie

Q4. Dynamic programming: min cost path in grid moving right/down:

- A) DP table
- B) Greedy always optimal
- C) Stack only
- D) Queue only

Q5. Union of rider/driver matching components:

- A) Union-Find
- B) Binary search
- C) Quick sort
- D) KMP

Q6. Topological order of dependent microservices:

- A) Kahn / DFS on DAG
- B) BFS on undirected cycle
- C) Heap sort
- D) Selection

Q7. Sliding window longest substring without repeat:

- A) Two pointers + set
- B) Dijkstra
- C) Prim
- D) Bellman

Q8. Heap of size k for k nearest drivers:

- A) Max-heap of size k / priority queue
- B) Stack
- C) Queue FIFO only
- D) Array unsorted only way

Q9. Detect negative cycle in pricing graph:

- A) Bellman-Ford
- B) Dijkstra
- C) BFS unweighted
- D) Binary search

Q10. Matrix spiral order traversal is:

- A) Simulation $O(mn)$
- B) Hash only
- C) Dijkstra
- D) Trie

Technical MCQ

Q11. Geo-hash / tiling helps:

- A) Spatial indexing of locations
- B) CSS grid only
- C) SMTP
- D) FTP

Q12. Pub/Sub for trip events enables:

- A) Async fan-out to consumers
- B) Sync only SQL join
- C) Only HTML
- D) Only DNS

Q13. Consistent hashing helps:

- A) Minimize remaps when nodes added
- B) Sort arrays
- C) Compile C
- D) Paint UI

Q14. ETA model serving latency cared via:

- A) Caching / approximate indexes
- B) Larger fonts
- C) More comments
- D) Slower disks only

Q15. Idempotent complete-trip API avoids:

- A) Double charge / double complete
- B) Faster GC
- C) More RAM
- D) TLS

Quantitative Ability

Q16. Speed 40 km/h, time 45 min. Distance:

- A) 30 km
- B) 20
- C) 40
- D) 25

Q17. Fare base 50 + 12/km for 8 km:

- A) 146
- B) 140
- C) 150
- D) 130

Q18. Two trains 60 & 90 km/h toward, 150 km apart. Meet after:

- A) 1 hour
- B) 1.5
- C) 2
- D) 0.5

Logical Reasoning

Q19. Series: 2, 3, 5, 8, 12, ?

- A) 17
- B) 15
- C) 16
- D) 18

Q20. All cars are vehicles. Some vehicles are bikes. Some cars are bikes?

- A) Does not follow
- B) Follows
- C) Either
- D) Only

Verbal Ability

Q21. Synonym of 'dispatch':

- A) send out
- B) delete
- C) compile
- D) encrypt

Q22. Antonym of 'scarce' capacity:

- A) abundant
- B) rare
- C) limited
- D) sparse

Programming Logic

Q23. A* search adds to Dijkstra:

- A) Heuristic estimate
- B) Only DFS
- C) Only sorting
- D) Only hashing

Q24. Interval scheduling max non-overlapping:

- A) Sort by end time greedy
- B) DFS
- C) Dijkstra
- D) KMP

Technical MCQ

Q25. Circuit breaker in rider API gateway:

- A) Stops cascading outages
- B) Increases thrashing
- C) Disables TLS
- D) Removes indexes

Q26. CQRS separates:

- A) Read and write models
- B) HTML and CSS only
- C) TCP and UDP only
- D) Stack and heap only

Programming Logic

Q27. Count islands in binary grid:

- A) DFS/BFS flood fill
- B) Binary search
- C) Quick sort
- D) Dijkstra

Answer Key & Explanations

Programming Logic

Q1. A
Dijkstra.

Q2. A
BFS hops.

Q3. A
Sweep.

Q4. A
Grid DP.

Q5. A
DSU.

Q6. A
Topo.

Q7. A
Window.

Q8. A
Heap.

Q9. A
BF.

Q10. A
Sim.

Technical MCQ

Q11. A
Geo.

Q12. A
Pubsub.

Q13. A
CH.

Q14. A
Latency.

Q15. A
Idempotent.

Quantitative Ability

Q16. A
 40×0.75 .

Q17. A
 $50 + 96$.

Q18. A
 $150/150$.

Logical Reasoning

Q19. A
 $+1+2+3+4+5$.

Q20. A
No.

Verbal Ability

Q21. A
Send.

Q22. A
Abundant.

Programming Logic

Q23. A
Heuristic.

Q24. A
Greedy ends.

Technical MCQ

Q25. A
CB.

Q26. A
CQRS.

Programming Logic

Q27. A
Flood fill.