1. What is the base case of recursion in a problem involving traversing a matrix?
   1. Reached a corner of the matrix
   2. **Reached outside the matrix boundaries**
   3. All elements of the matrix have been visited
   4. The current element is zero
2. In a maze-solving problem using recursion, what does a return value of `true` indicate?
   1. Maze has no solution
   2. **Reached the destination**
   3. Stuck in a dead-end
   4. Recursion stack overflow
3. When implementing flood-fill using recursion, what data structure is commonly used to keep track of visited cells?
   1. Queue
   2. Stack
   3. **Set**
   4. Array
4. What is the time complexity of recursively finding all paths from the top-left corner to the bottom-right corner of an NxM matrix?
   1. O(N)
   2. O(M)
   3. O(N+M)
   4. **O(2^(N+M))**
5. In dynamic programming, recursion with memoization is often used to optimize problems involving grids. What is memoization?
   1. Keeping a record of function calls and their parameters
   2. Using matrices instead of grids for better performance
   3. **Storing computed values to avoid redundant calculations**
   4. Measuring memory usage during recursion
6. Which type of recursion involves a function calling itself multiple times for smaller sub-problems?
   1. Tail recursion
   2. Linear recursion
   3. Binary recursion
   4. **Tree recursion**
7. What is the drawback of using pure recursion to calculate Fibonacci numbers?
   1. It's difficult to implement
   2. **It's slow due to redundant calculations**
   3. It requires complex mathematical formulas
   4. It can only calculate small Fibonacci numbers
8. In a problem involving backtracking on a matrix, what is backtracking?
   1. Trying all possible paths through the matrix
   2. Going backward in the matrix
   3. **Undoing the last move and trying a different path**
   4. Starting from the end of the matrix and working backwards
9. When solving a Sudoku puzzle recursively, what is the role of constraints?
   1. To ensure the puzzle has a unique solution
   2. **To limit the possibilities for each cell**
   3. To restrict the use of recursion
   4. To reduce the size of the matrix
10. In a problem involving finding a path from a source to a destination in a matrix, what is the purpose of marking visited cells?
    1. **To avoid revisiting cells and getting stuck in a loop**
    2. To improve the visual representation of the matrix
    3. To increase the runtime efficiency of the algorithm
    4. To indicate that the path has been found
11. What is the base case for a recursive algorithm to calculate the sum of elements in a matrix?
    1. Reached a corner of the matrix
    2. **Reached outside the matrix boundaries**
    3. All elements of the matrix have been visited
    4. The current element is zero
12. When using memoization in a recursive algorithm on a grid, where are the computed results stored?
    1. In an auxiliary grid
    2. **In a hash table**
    3. In a stack
    4. In an array
13. What is the time complexity of traversing an NxM matrix recursively, visiting each cell once?
    1. O(N)
    2. O(M)
    3. O(N+M)
    4. **O(N\*M)**
14. In a matrix, how are rows and columns typically represented using recursion?
    1. As linked lists
    2. **As arrays**
    3. As trees
    4. As queues
15. In a recursive function, what is the purpose of passing the current position as a parameter?
    1. To keep track of the total number of elements visited
    2. To determine if the destination has been reached
    3. To avoid redundant calculations
    4. **To know where to start the next recursive call**
16. What type of recursion involves a function calling itself with a modified parameter?
    1. Indirect recursion
    2. Tail recursion
    3. Helper recursion
    4. **Direct recursion**
17. When implementing a problem involving movement on a grid, what can be done to prevent going out of bounds?
    1. Use a try-catch block
    2. **Check the boundaries before making a move**
    3. Change the starting position
    4. Use a bigger grid
18. In a maze-solving problem using recursion, what does a return value of `false` indicate?
    1. Maze has no solution
    2. Reached the destination
    3. **Stuck in a dead-end**
    4. Recursion stack overflow
19. What is the time complexity of finding the factorial of a number using recursion?
    1. O(1)
    2. **O(n)**
    3. O(log n)
    4. O(n^2)
20. In dynamic programming, what is the top-down approach?
    1. Starting from the bottom of the grid and working upwards
    2. Solving larger sub-problems before smaller ones
    3. Breaking the problem into smaller sub-problems
    4. **Starting with the initial problem and breaking it down into simpler cases**
21. What is the purpose of a visited array in recursive algorithms on a grid?
    1. To store the values of cells that have been visited
    2. To mark cells that are unreachable
    3. **To avoid revisiting cells and getting stuck in a loop**
    4. To indicate the starting point of the recursion
22. In a problem involving finding connected components in a grid, what is a connected component?
    1. **A set of cells that are adjacent to each other**
    2. A subset of the matrix
    3. A sequence of cells that can be traversed linearly
    4. A group of cells that are the same color
23. What is the purpose of a base case in a recursive algorithm on a matrix?
    1. To mark the starting point of the recursion
    2. **To handle the smallest sub-problems directly**
    3. To store the computed values for memoization
    4. To keep track of visited cells
24. In a maze-solving problem using recursive backtracking, what does the term "backtracking" refer to?
    1. Going back to the beginning of the maze
    2. Revisiting cells that have already been visited
    3. **Undoing a move and exploring a different path**
    4. Tracing the steps taken in the maze
25. What is the role of recursion in problems that involve permutation of elements in a grid?
    1. To ensure that all permutations are unique
    2. To create a new grid with permuted elements
    3. **To generate all possible permutations**
    4. To sort the elements in the grid
26. In a problem involving finding the shortest path in a grid, how can dynamic programming be used with recursion?
    1. By calculating the shortest path between each pair of cells
    2. **By recursively calculating the shortest path from the start to each cell**
    3. By storing the computed distances in a matrix and reusing them
    4. By recursively dividing the grid into smaller sub-grids
27. What is the key characteristic of a tail-recursive function?
    1. It calls itself multiple times for smaller sub-problems
    2. It uses a base case to terminate the recursion
    3. **It performs a single recursive call at the end of the function**
    4. It uses memoization to optimize the recursion
28. In a maze-solving problem, what is the role of recursion?
    1. To keep track of the visited cells
    2. **To backtrack and explore different paths**
    3. To optimize the path-finding algorithm
    4. To create a new maze configuration
29. What is the primary reason for using recursion in matrix traversal problems?
    1. It's easier to implement than iterative solutions
    2. It leads to faster execution times
    3. It simplifies the problem by avoiding complex data structures
    4. **It provides a natural way to handle backtracking and sub-problems**
30. In a problem involving counting paths in a matrix, what does the recursive formula for calculating paths typically involve?
    1. **Addition of the number of paths from the left and right**
    2. Multiplication of the number of paths from the left and right
    3. Division of the number of paths from the left by the number from the right
    4. Subtraction of the number of paths from the left and right
31. When implementing a problem involving movement on a grid, how can you ensure that each cell is visited exactly once?
    1. **Use a visited array**
    2. Use a counter variable
    3. Visit cells randomly
    4. Visit cells in a spiral pattern
32. In a recursive function that traverses a grid, what should be the criteria for making a recursive call?
    1. Moving to the cell with the smallest value
    2. **Moving to the unvisited neighboring cell**
    3. Moving to the cell with the largest value
    4. Moving to a random neighboring cell