

Different Programming Techniques:-

1. **Recursion**: A function calls itself to solve a problem by breaking it down into smaller subproblems.
2. **Backtracking**: A systematic trial-and-error approach to solve problems by making choices and undoing them if they lead to an incorrect solution.
3. **Greedy Algorithms**: Make locally optimal choices at each step to find a global optimum.
4. **Dynamic Programming**: Solve problems by breaking them down into smaller subproblems and storing solutions to avoid redundant calculations.
5. **Divide and Conquer**: Divide a problem into smaller, similar subproblems, solve them, and combine their solutions.
6. **Branch and Bound**: An algorithmic technique for optimization problems, often used with backtracking.
7. **Branch and Bound**: An algorithmic technique for optimization problems, often used with backtracking.
8. **Heuristic Algorithms**: Use approximation techniques to find a near-optimal solution when an exact solution is impractical.
9. **Randomised Algorithms**: Make use of randomness to solve problems or find approximate solutions.
10. **Parallel and Concurrent Programming**: Design software to execute tasks simultaneously to improve performance.
11. **Memoization**: Caching previously computed results to avoid redundant calculations.
12. **Data Compression Algorithms**: Reduce the size of data for storage or transmission.
13. **Pattern Matching Algorithms**: Find occurrences of a pattern within a larger text or data set.
14. **Graph Algorithms**: Algorithms for working with graphs, such as breadth-first search (BFS) and depth-first search (DFS).
15. **String Algorithms**: Algorithms for string manipulation and pattern searching, like the Knuth-Morris-Pratt algorithm.
16. **Numerical Algorithms**: Algorithms for numerical computations, such as root-finding algorithms and numerical integration.