SPPU-BE-COMP-CONTENT – KSKA Git

DAA

ASSIGNMENT - 5

Q1

Backtracking:

→ Systematic algorithmic approach for solving constraint satisfaction problems. It tries to build a solution incrementally & abandons a path as soon as it determines that it cannot lead to a valid solution.

Peculiar Characteristics:

→ Explores all possible solutions in a state space tree
→ Avoids exploring paths that violate constraints (pruning
→ Can be implemented recursively or iteratively.
→ Efficient for problems with many possible combin's but fewer valid solutions

Applications:

- N-Queens problem
- Sudoku solver
- Maze / path finding problems
- Graph colouring

Q2

Explicit constraints: Directly defined rules for the probl

- No two queens can be in same row
- No two queens can be in same column
- No two queens can be in same diagonal

Implicit constraints: constraints that are automatically satisfied by algorithm design

- Placing one queen per row avoids row conflicts automatically.
- Only valid column choices are considered during placement.

Q3

ASPECT	RECURSIVE BACKTRACKING	NON-RECURSIVE BACKTRACKING
Time complexity	O(n!) (all permutations may be tried)	O(n!) (similar exploration but explicit stack used)
Space complexity	O(n) for recursion stack	O(n) for explicit stack
Ease of implemⁿ	Simpler, cleaner	Slightly more complex needs stack management
Debugging	Easier to trace	Harder to trace

Q4 APPLICATIONS

1. Puzzle Solvers:
Automating solutions for chess, puzzles, Sudoku or crosswords.

2. Optimization problems: Resource allocation, scheduling or assignment problems where constraints must be satisfied.

3 AI / Game theory: Path planning, decision making in games where multiple combin^n are possible.