Data–Driven Crop Recommendation System Using Dynamic Environmental Inputs

B.Tech (AIDS)
UG-II (III-Semester)

Subject Coordinator
Dr. Vijay Kumar Soni(Assistant Professor)

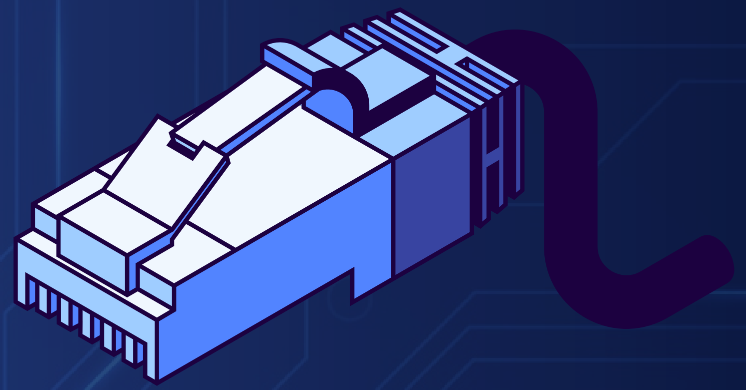
श्रद्धावान् लभते ज्ञानम्

Presented By:
DHWANIT SHARMA (DL.AI.U4AID24111)
KANAN GOEL (DL.AI.U4AID24017)
PARKHI MAHAJAN (DL.AI.U4AID24126)
PUNIT LOHAN (DL.AI.U4AID24127)

AMRITA VISHWA VIDYAPEETHAM, DELHI NCR, FARIDABAD
Department of Artificial Intelligence & Data Science (AIDS)

MOTIVATION

- Agriculture is highly dependent on rainfall.
- Unexpected rainfall disrupts crop planning.
- Manual crop selection is slow and subjective.
- Farmers need fast, reliable, and optimal decisions.
- This project focuses on algorithmic decision-making.



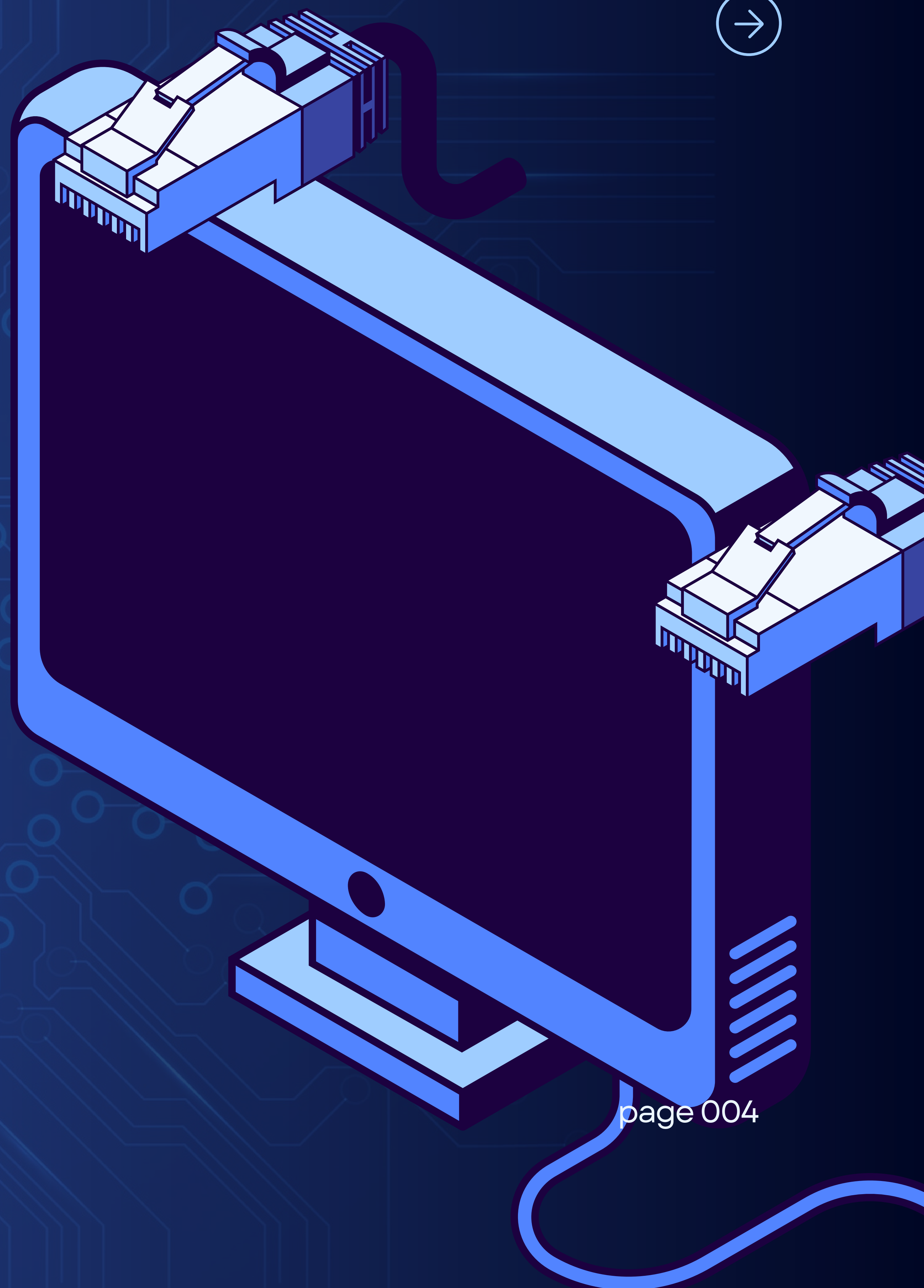
PROBLEM SCENARIO

- Agriculture is sensitive to rainfall variation.
- Sudden rainfall changes water availability.
- Manual crop selection is slow and suboptimal.
- Wrong selection can lead to economic loss, creating the need for fast, explainable decision support.

GOAL:
Assist crop selection under limited water conditions.

PROBLEM **STATEMENT**

Given a set of crops with known water requirements and profits, and a fixed amount of available water, determine the optimal subset of crops that maximizes total profit without exceeding the water constraint.



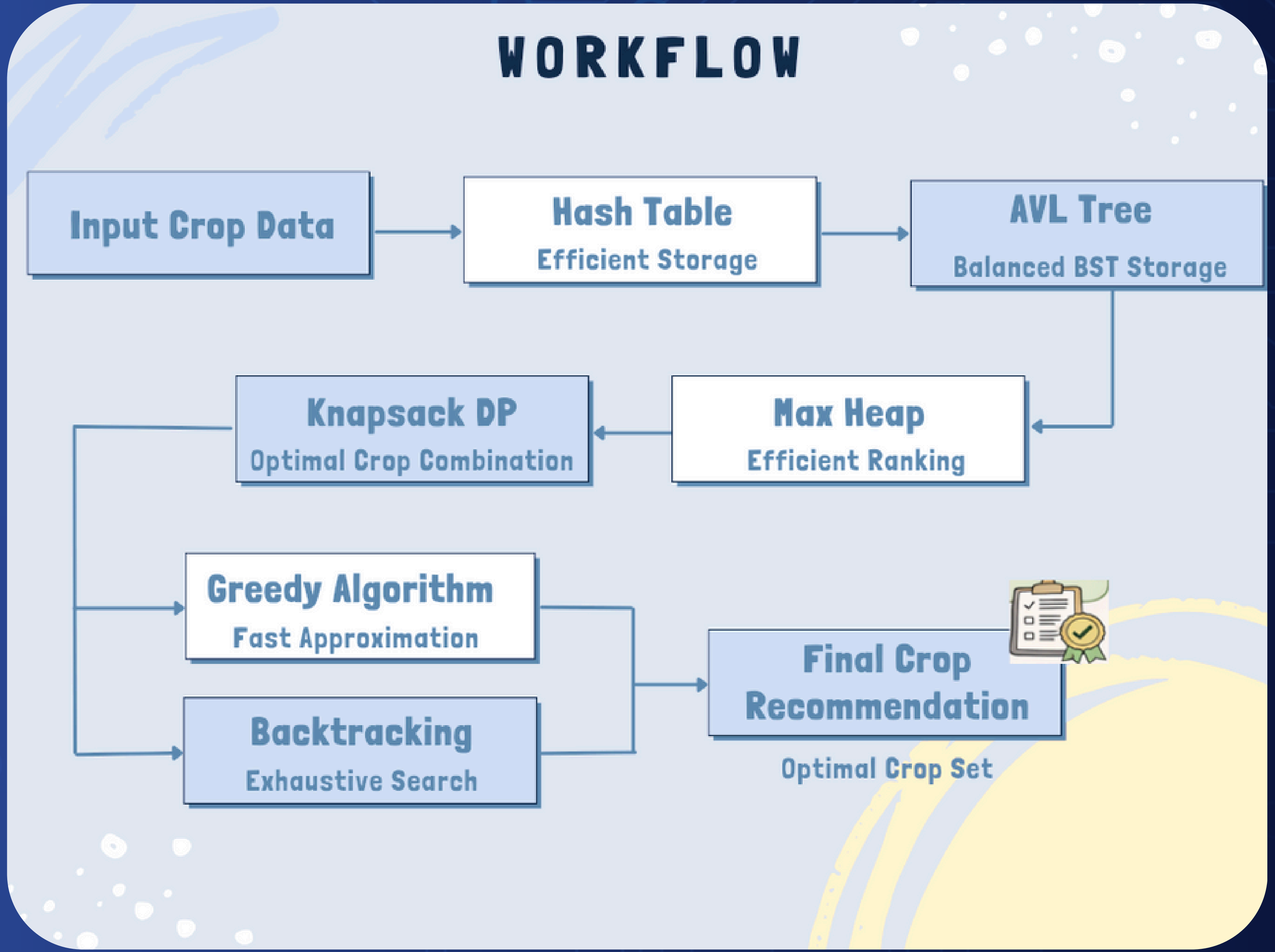
EXISTING TECHNOLOGIES

AND THEIR LIMITATIONS

- ML models → black box, data-hungry
- Rule-based systems → rigid
- Manual decisions → subjective
- Sorting-based tools → no constraint handling

PROPOSED **SYSTEM OVERVIEW**

- Deterministic, explainable system.
- Uses classical data structures & algorithms.
- No machine learning.
- Console-based Java implementation.



SYSTEM **WORKFLOW**

1. Crop data input
2. Efficient storage & retrieval
3. Crop ranking
4. Optimal selection under water constraint
5. Result validation

DATA **MODELING**

Each crop contains:

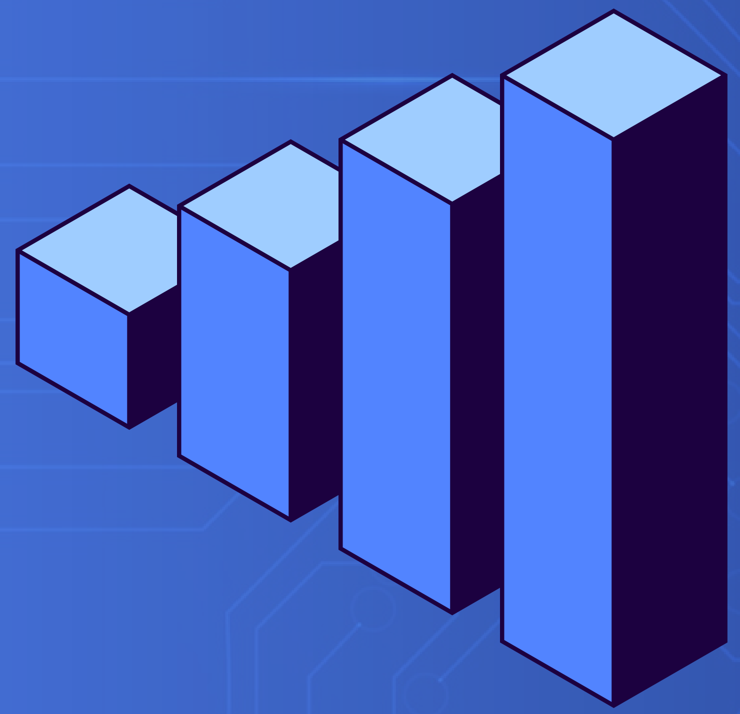
- Crop name
- Water requirement (weight)
- Expected profit (value)

DATA **STRUCTURES USED**

DATA STRUCTURE	FUNCTIONAL ROLE	DESIGN RATIONALE
Linked List	Hash collision resolution	Flexible memory and chaining
Hash Table	Data retrieval layer	O(1) average access time
Binary Search Tree	Ordered baseline	Highlights imbalance limitations
AVL Tree	Balanced ordering	Guaranteed logarithmic operations
Max Heap	Priority ranking	Efficient max-element access

LINKED **LIST**

- Dynamic linear data structure.
- Efficient insertions without memory reallocation.
- Commonly used for collision handling in hash tables.



HASH **TABLE**

- Key-value based access structure.
- Enables average-case constant-time lookup.
- Performance depends on hash function and collision strategy.

BINARY **SEARCH TREE**

Hierarchical structure maintaining sorted order.
Enables ordered traversal of elements.
Performance degrades if tree becomes skewed.

AVL **TREE**

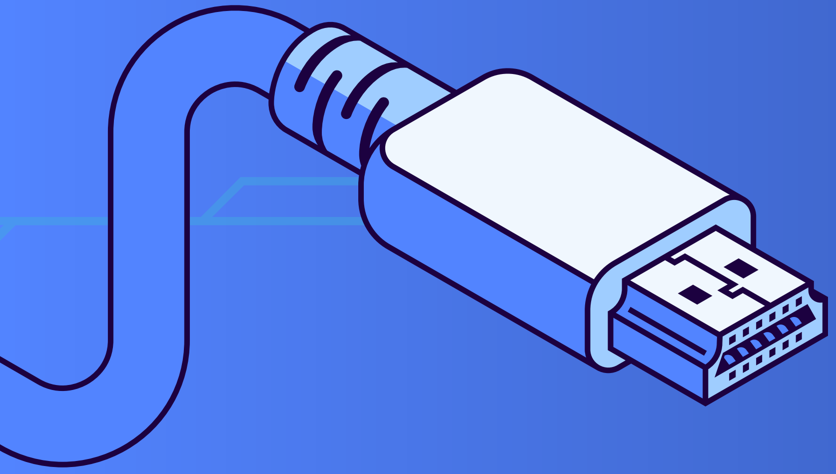
- Self-balancing binary search tree.
- Maintains height balance using rotations.
- Guarantees logarithmic search and insertion.

MAX **HEAP**

Complete binary tree with priority property.
Root always contains the maximum element.
Optimized for repeated max-element access.

ALGORITHMS **IMPLEMENTED**

ALGORITHM	FUNCTIONAL ROLE	DESIGN RATIONALE
Dynamic Programming (0/1 Knapsack)	Optimal crop selection	Guarantees global optimality
Greedy Algorithm	Quick approximation	Fast but locally optimal
Backtracking	Result validation	Exhaustive correctness check
Branch & Bound	Optimized validation	Prunes unnecessary searches

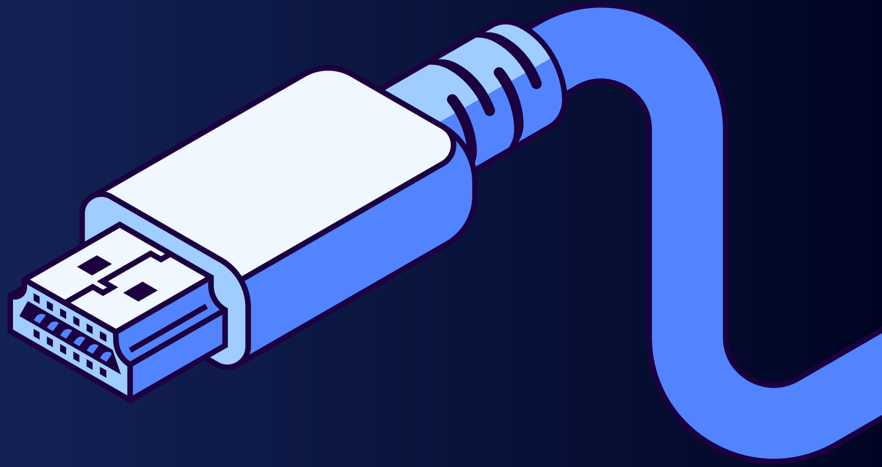


DYNAMIC **PROGRAMMING**

Breaks problem into overlapping subproblems.
Stores intermediate results to avoid recomputation.
Suitable for optimization under constraints.

GREEDY **ALGORITHM**

Makes locally optimal decisions at each step.
Simple and fast to implement.
Does not reconsider past choices.



BACKTRACKING **ALGORITHM**

- Explores all possible solution paths recursively.
- Rejects partial solutions that violate constraints.
- Guarantees correctness.

BRANCH **AND BOUND**

Optimization over backtracking.
Uses bounds to eliminate non-promising paths.
Reduces search space without losing optimality.

SYSTEM **OUTPUT**

```
--- AVL Tree (Sorted Crops) ---
Maize Profit: 65
Wheat Profit: 70
Rice Profit: 90
Sugarcane Profit: 120
```

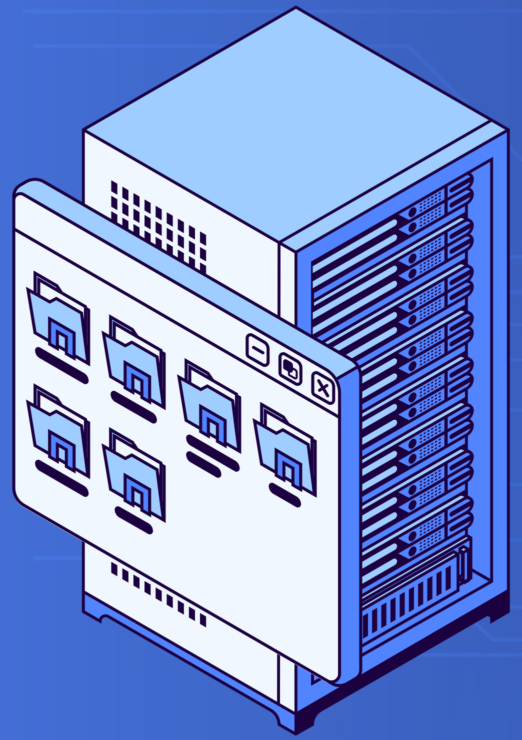
AVL Tree output showing crops sorted by expected profit **01.**

02. **Max Heap output identifying the highest-profit crop**

```
--- Max Heap (Top Crops) ---
Best Crop: Sugarcane
```

```
--- Knapsack (DP) ---
Max Profit: 185
```

Dynamic Programming output showing maximum achievable profit **03.**



```
--- Greedy Selection ---
Quick Choice: Maize
```

04. Greedy algorithm output showing locally optimal crop selection

```
--- Backtracking Simulation ---
Best Profit (Backtracking): 185
```

Backtracking output validating optimal profit 05.

OBSERVATIONS

- Crops are correctly sorted using AVL Tree traversal.
- Heap efficiently identifies the highest-profit crop.
- Greedy algorithm selects a locally optimal crop.
- Dynamic Programming computes the maximum profit under constraints.
- Backtracking result matches Dynamic Programming output.

INFERENCE

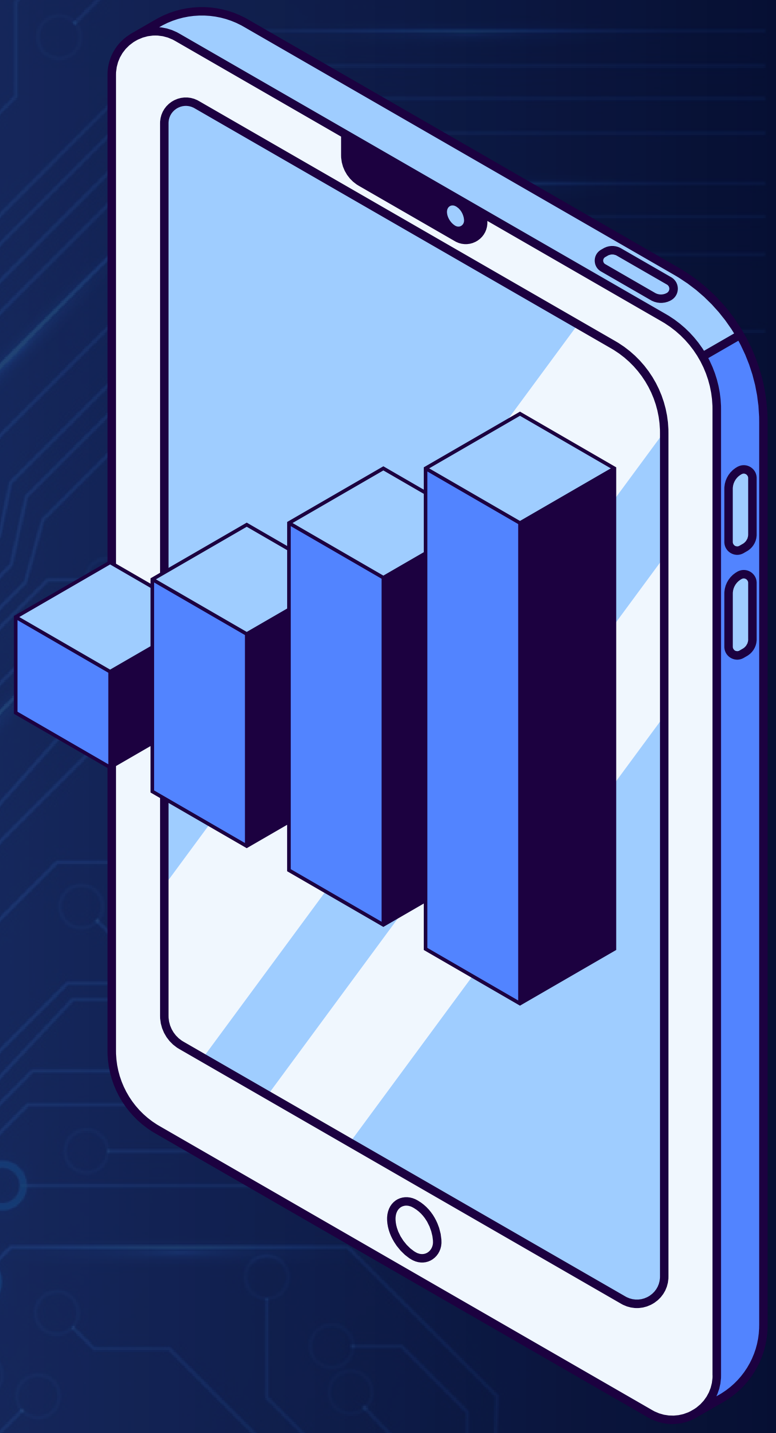
- Data structures support ordering and prioritization but not optimization.
- Greedy approach is fast but does not guarantee optimality.
- Dynamic Programming provides the globally optimal solution.
- Backtracking confirms correctness of the optimal result.

KEY **FINDINGS**

- Dynamic Programming achieves global optimality under water constraints — 01.
- Greedy approach demonstrates limitation of local decision-making — 02.
- Backtracking validates the correctness of the optimal solution — 03.
- Data structures support efficient ordering and prioritization — 04.
- System executes successfully without errors — 05.

CONCLUSION

- Successfully modeled crop selection as a constrained optimization problem.
- Demonstrated practical use of classical data structures for data organization.
- Applied Dynamic Programming to achieve globally optimal crop selection.
- Validated optimality using exhaustive backtracking.
- Highlighted limitations of greedy strategies in constrained problems.
- Ensured deterministic and explainable decision-making.
- System performs efficiently within the defined scope.



FUTURE **SCOPES**

- Integration with real-time weather APIs for dynamic water updates.
- GUI-based interface for improved farmer usability.
- Inclusion of soil nutrient and fertility parameters.
- Support for multi-season and long-term crop planning.

THANK

YOU.