CASE STUDY 1: SOLVING REAL WORLD PROBLEM

" Application of Dynamic Programming and/or "

Graph in Reservoir Management

PRESENTED BY:

Cabacungan, John Martin

De los Reyes, Jann Moises Nyll

Quibral, Julian Vincent

CPE22S3

Problem:

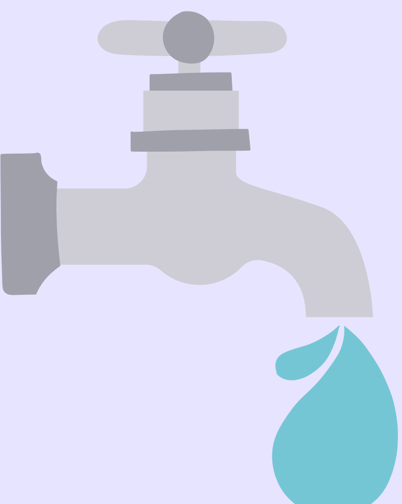
Efficiently operating reservoirs involves making decisions about water release, storage, and inflow management. During flood seasons, reservoirs must balance the need for dam safety (releasing water) with minimizing downstream flood risk (releasing as little water as possible).

Objective:

To optimize reservoir releases by minimizing the maximum release from a reservoir to reduce flood risk.

Algorithm:

Dynamic programming helps manage reservoirs during flood events, balancing safety and downstream impact.



Iteration 1

Problem Identifcation

How can we optimize the reservoir operation to reduce flood risk downstream?

Decomposition

- inflow prediction
- release scheduling
- flood control

Pattern Recognition

- Pattern in historical inflow, outflow and flood events.
- identifying recurring pattern during flood, rainfall and dry season

Abstraction

- Relevant :flood risk reduction, energy optimization
- Trade-off : Flood control vs hydropower

Iteration 2

Problem Identifcation

We want to minimizing the maximum release from a reservoir to reduce flood risk.

Decomposition

- Reservoir Storage Level
- Inflow Prediction
- Release Decision points

Pattern Recognition

- Recognize patterns in past flood events, inflow variations, and their impact on downstream areas.
- Identify recurring patterns related to seasonal rainfall, , and extreme weather events.

Abstraction

- Focus on minimizing the maximum release while ensuring downstream safety.

CODE

```
[ ] '''
    Optimizing Reservoir Releases by Minimizing the Maximum release from a  reservoir

    Problem:
        Efficiently operating reservoirs involves making decisions about water release,
    storage, and inflow management. During flood seasons, reservoirs must balance
    the need for dam safety (releasing water) with minimizing downstream flood
    risk (releasing as little water as possible).

    Things to note:
    - Choose the best optimal release
    Approach:
    1.) Create a function that determines the release policy to maximize the total inflow
    2.) It should be display the optimal release
    '''
```

```python
def find_optimal_release(inflow, weights, k):
    n = len(inflow)
    dp = [[0] * (k + 1) for _ in range(n + 1)]

    # Initialize the DP table
    for i in range(1, n + 1):
        for j in range(1, k + 1):
            dp[i][j] = max(dp[i - 1][j], dp[i - 1][j - weights[i - 1]] + inflow[i - 1])

    # Backtrack to find the optimal release trajectory
    solution = []
    i = n
    j = k
    while j > 0 and i > 0:
        if dp[i][j] != dp[i - 1][j]:
            solution.append(i)
            j -= weights[i - 1]
        i -= 1
    return solution
```

TESTING

```
# Test :
inflow = [10, 15, 20, 25, 30, 35, 40]  # Example inflow data (in cubic meter)
weights = [2, 3, 4, 5, 6, 7, 8]      # Example weights (release amounts in cubic meter)
k = 15                       # Maximum allowable release

optimal_release = find_optimal_release(inflow, weights, k)
print("Optimal release indices:", optimal_release)
```

```
Optimal release indices: [7, 5, 4]
```