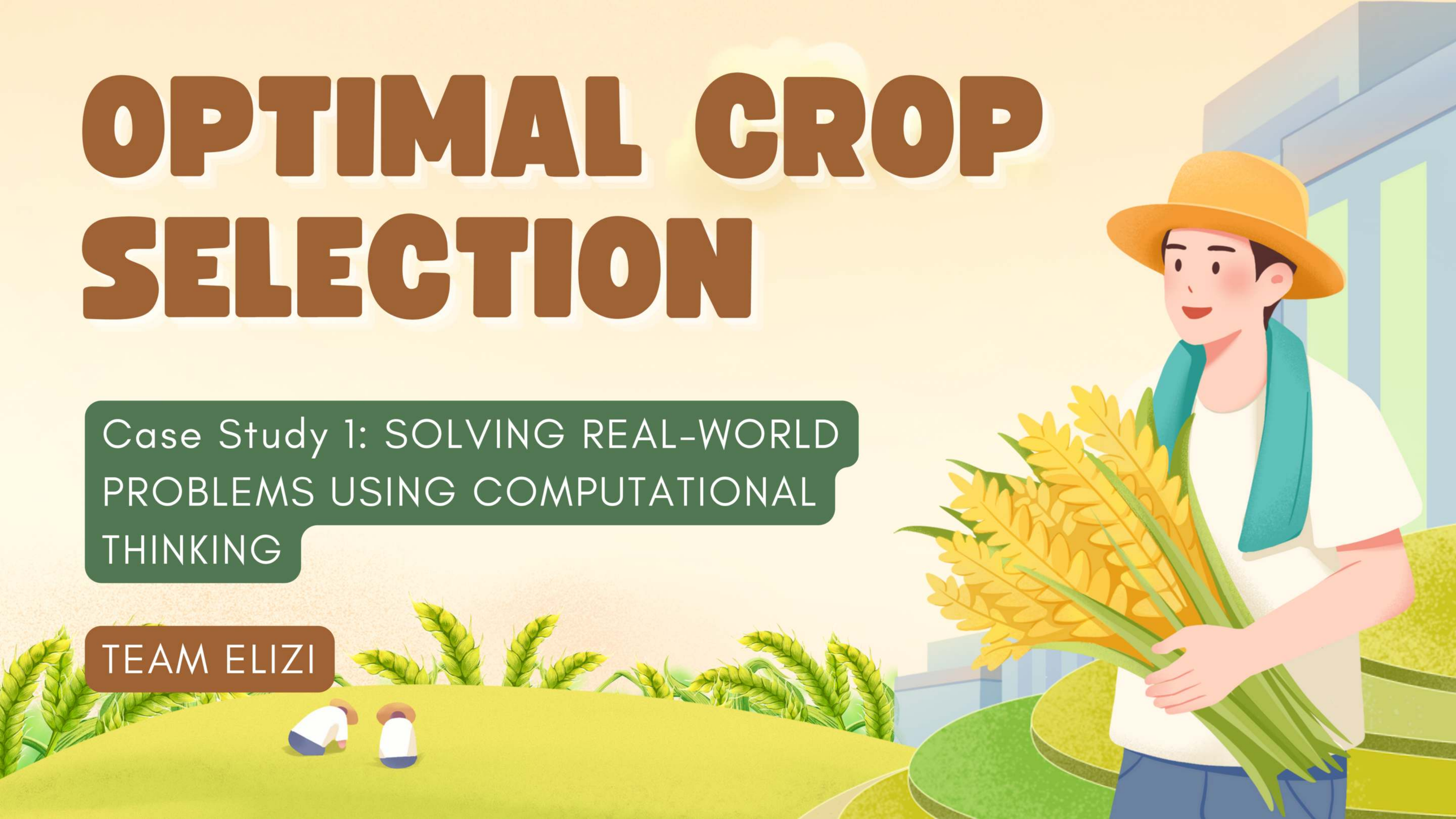
OPTIMAL CROP SELECTION

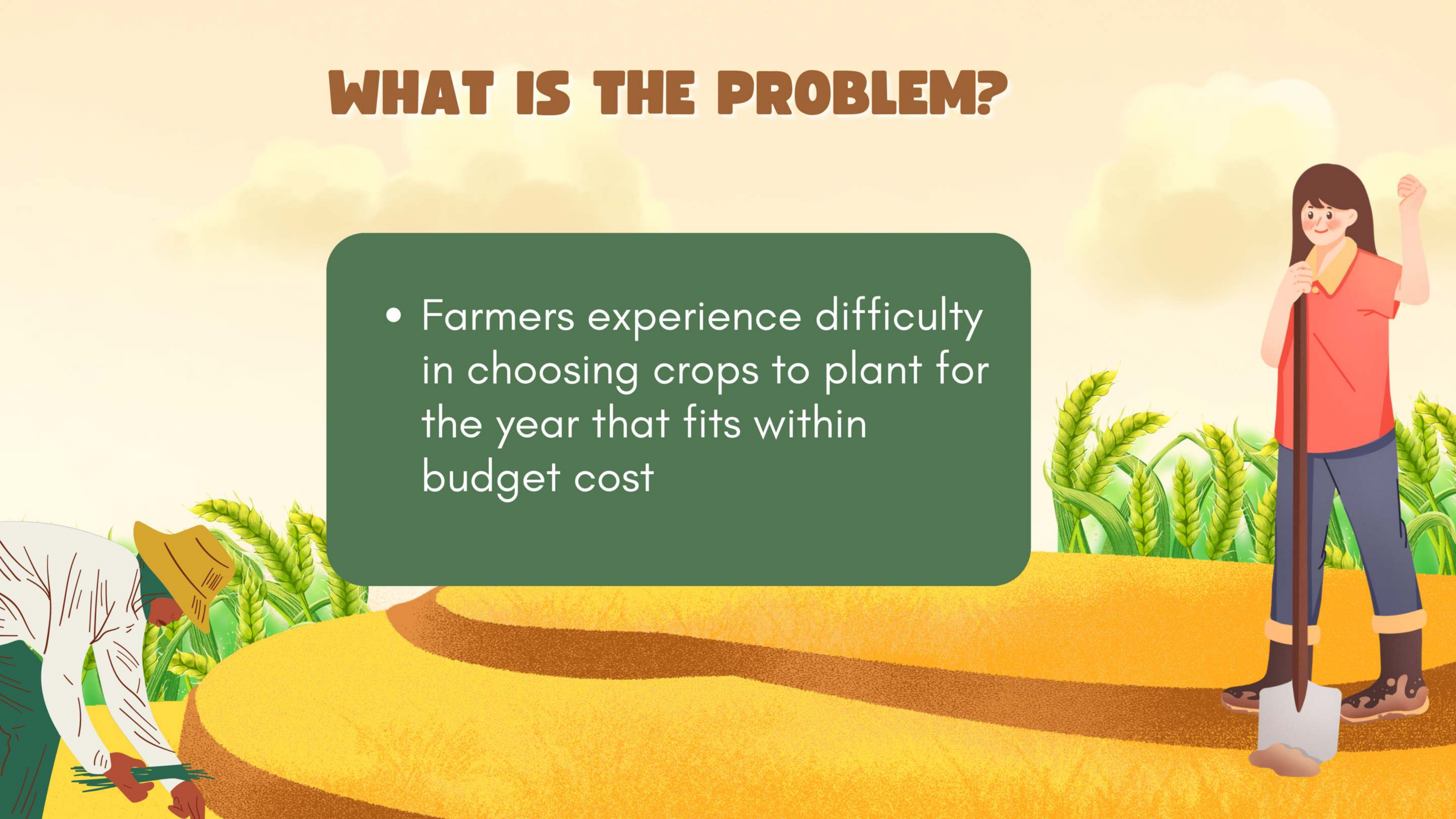
Case Study 1: SOLVING REAL-WORLD PROBLEMS USING COMPUTATIONAL THINKING

TEAM ELIZI



WHAT IS THE PROBLEM?

- Farmers experience difficulty in choosing crops to plant for the year that fits within budget cost



1ST ITERATION

What are the crops for each soil type

DECOMPOSITION

- List it all types of soil
- List the crops
- List crops price

PATTERN RECOGNITION

- Certain crops may have similar required soil type with other crops

ABSTRACTION

- Only relevant information is soil type and crops, and prices.
- Soil nutrients contents are irrelevant.

2ND ITERATION

Problem:
What affects probable crop yield?

DECOMPOSITION

- List the temperature requirement per crop
- List the humidity requirement per crop

PATTERN RECOGNITION

- Crops may have the same temperature requirements
- Crops may have the same relative humidity requirements

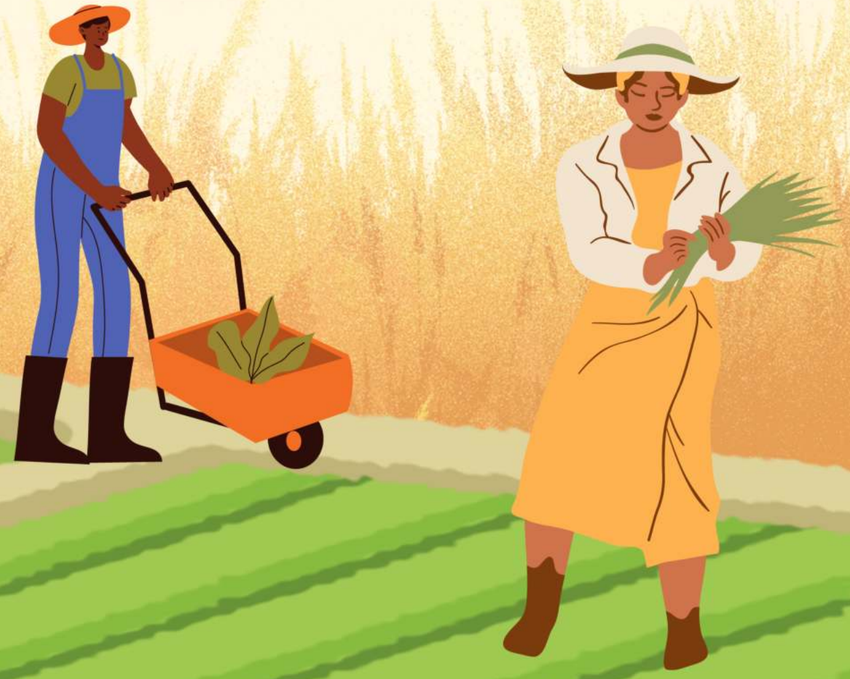
ABSTRACTION

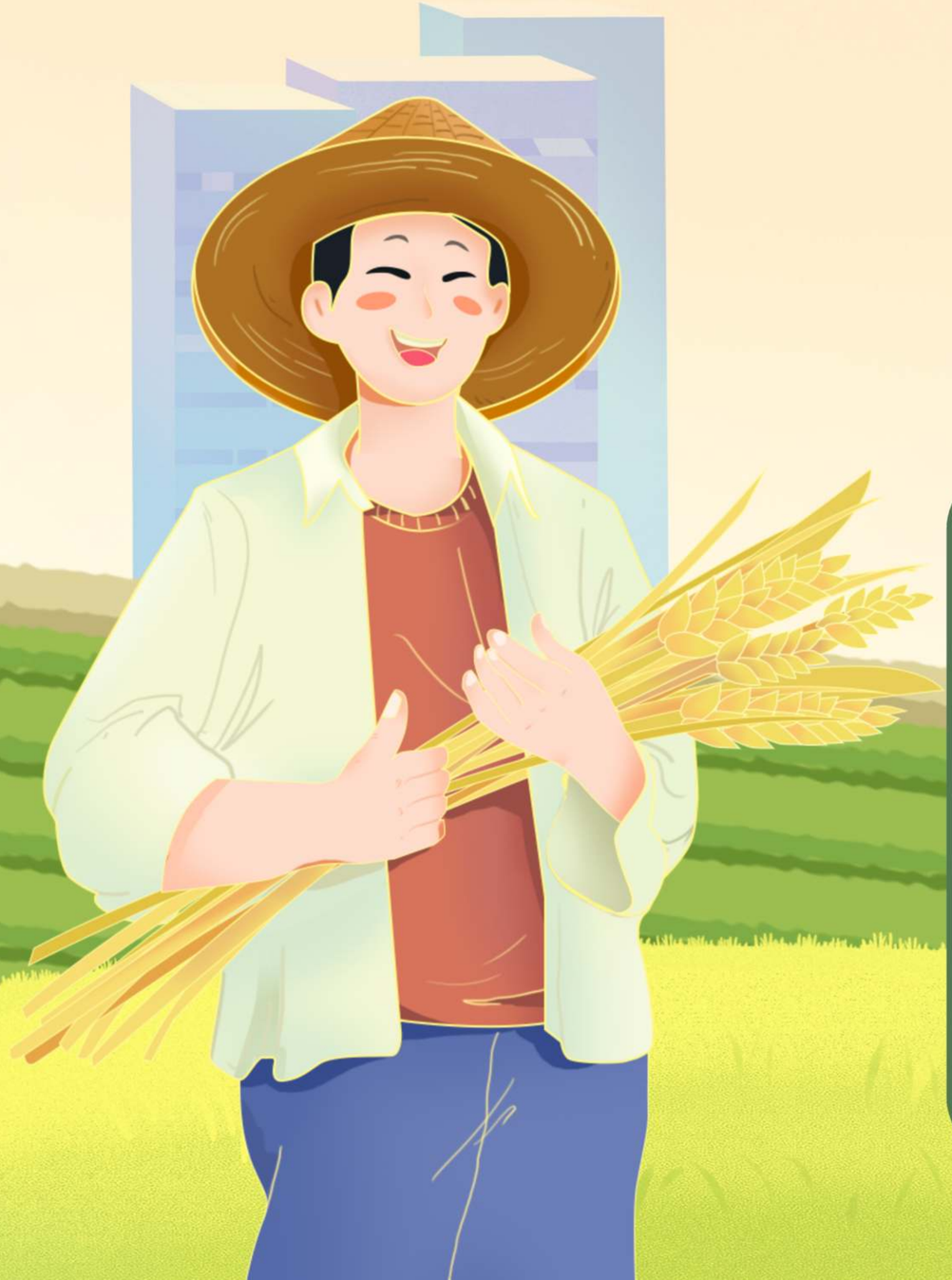
- Getting the two parameters will help in finding the best crop

3RD ITERATION

What is the best crop for a quarterly time frame fit within the budget?

Decomposition	• Analyze quarterly changes in temperature and humidity. • Determine suitable crops for each quarter based on these parameters. • Determine if crop is within the budget
Pattern Regconition	• Some crops may be applicable for many quarters when in range of the parameters
Abstraction	• Climate parameters in quarterly format are deemed important • Crop growth duration is irrelevant • Possible profit yield is irrelevant



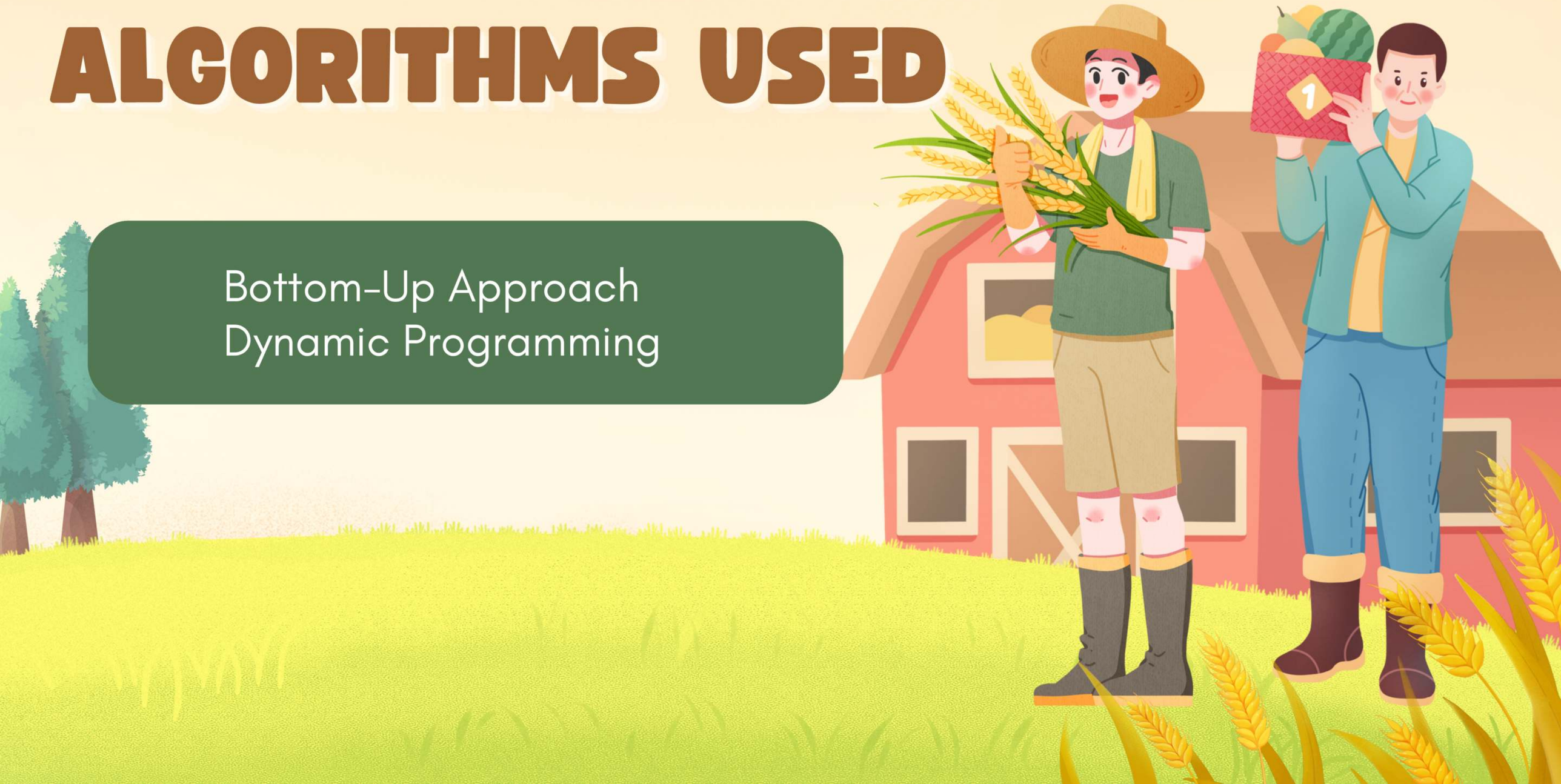


PROPOSED SOLUTION

Our proposed solution uses a program that takes inputs about crops, soil type, crop cost budget, a quarterly record of minimum and maximum temperature and humidity which is used to be parameters to check which crop is the possible to be planted in these conditions. Then we will check if those crops are still within the given budget. The output will give the optimal combination of crops to be planted in the quarter.

ALGORITHMS USED

Bottom-Up Approach
Dynamic Programming



CODE BREAK DOWN

```python
import pandas as pd

def DP_TABULATION(Names, temperature, Humidity, Soils, Prices, max_hum, min_hum, max_temp, min_temp, GIVEN_BUDGET):
    rows = len(Prices)

    # Initialize DP table
    # instead na 0 gawin empty list kasi mag base tayo sa len ng crops pag dating sa pag kokompara
    table = [[[] for _ in range(GIVEN_BUDGET + 1)] for _ in range(rows + 1)]


    for row in range(1, rows + 1):
        for column in range(1, GIVEN_BUDGET + 1):
          #initialize the next value to avoid redundunt
            crop_name = Names[row - 1]
            crop_temp = temperature[row - 1]
            crop_hum = Humidity[row - 1]
            crop_soil = Soils[row - 1]
            crop_price = Prices[row - 1]

            if crop_price <= column and (min_temp <= crop_temp <= max_temp) and (min_hum <= crop_hum <= max_hum):
                # Maximize by choosing the best crop combination
                # by checking who has the highest length of list of crops between previous row and the computed coordinates
                table[row][column] = max(table[row - 1][column],
                                         [(crop_name, crop_soil)] + table[row - 1][column - crop_price],
                                         key=len)
            else:
              # if not get the list crops of the previous row
                table[row][column] = table[row - 1][column]

    # Get the best crop combination
    best_combination = table[rows][GIVEN_BUDGET]

    # Organize results by soil type
    soil_dict = {}
    for crop, soil in best_combination:
        if soil not in soil_dict:
            soil_dict[soil] = []
        soil_dict[soil].append(crop)

    return soil_dict
```

CODE BREAK DOWN

```
def test_tabulation(Names, temperature, Humidity, Soils, Prices, GIVEN_HUMIDITY, GIVEN_TEMP, GIVEN_BUDGET):
    results = {}
    for quarter in range(4):

      results[f'Q{quarter+1}'] = DP_TABULATION(
            Names,
            temperature,
            Humidity,
            Soils,
            Prices,
            GIVEN_HUMIDITY[quarter][1],
            GIVEN_HUMIDITY[quarter][0],
            GIVEN_TEMP[quarter][1],
            GIVEN_TEMP[quarter][0],
            GIVEN_BUDGET
        )

    return results
```

CODE BREAK DOWN

```
dataframes = pd.read_csv('/content/UPDATED_CROPS.csv')
Names = dataframes['Crop Type']
temperature = dataframes['Temparature']
Humidity = dataframes['Humidity']
Soils = dataframes['Soil Type']
Prices = dataframes['Price(PHP)']

GIVEN_HUMIDITY = [(64,84),(50,70),(45,65),(60,80)]
GIVEN_TEMP = [(22,27),(25,36),(23,29),(22,27)]
GIVEN_BUDGET = 800


#here is the printing of results
for key, value in test_tabulation(Names, temperature, Humidity, Soils, Prices, GIVEN_HUMIDITY, GIVEN_TEMP, GIVEN_BUDGET).items():
  print(f'==============={key}====================')
  for keys,values in value.items():
    print(f'{keys}:{values}')

  print('\n')
```

SOURCES

- DEL PRADO, D. G. (2024, July 9). Annual Average Maximum Temperature Reached 31.3 degrees Celsius in 2023 | Philippine Statistics Authority | Republic of the Philippines. Psa.gov.ph. https://psa.gov.ph/content/annual-average-maximum-temperature-reached-313-degrees-celsius-2023

- Mapa, D. (2025). Authentication challenge pages. Psa.gov.ph. https://psa.gov.ph/statistics/environment/peenra/node/164736

- Wise. (2024, March 2). Philippines Climate: How Hot Is It Really? Pinay Wise. https://pinaywise.com/philippines-facts/philippines-climate-how-hot-is-it-really/

Q & A

Open floor for questions and discussion