**1) Programming Language**: Python

**2) Assumptions:**

* The input matrix is a list of lists without any irregularities such as mismatched row sizes or non-integer values.
* All stock prices are non-negative integers.
* The input matrix will not be empty and will contain at least one list with at least two price entries, to allow for a purchase and a sale.

**3) Definitions:**

* **matrix**: A list of lists in Python, where each sublist represents a stock and contains the prices of that stock for consecutive days.
* **max\_profit\_info**: A tuple that holds the best transaction details, including the stock index (1-indexed), the buy day (1-indexed), the sell day (1-indexed), and the maximum profit found.
* **stock\_index**: An integer representing the index of the current stock in the iteration, starting from 0 for the first stock.
* **prices**: A list of integers representing the prices of a stock over a series of days.
* **min\_price**: An integer representing the minimum stock price encountered so far for the current stock in the iteration.
* **max\_profit**: An integer representing the maximum profit found for a particular stock, initialized to 0 for each new stock in the iteration.
* **buy\_day** and **sell\_day**: Integers representing the days on which buying and selling would result in the **max\_profit**, respectively. Initialized to 1 (considering 1-indexing) and updated within the loop when a new minimum price is found or a higher profit is calculated.
* **current\_day**: An integer representing the index of the current day in the iteration, starting from 0 for the first day.
* **price**: An integer representing the current day's price of a stock.
* **current\_profit**: An integer calculated as the difference between the current **price** and **min\_price**, representing the potential profit if the stock were sold on the current day.

**4) Pseudocode:**

Function find\_max\_profit\_greedy\_approach takes a matrix of stock prices:

# Initialize a tuple to store the maximum profit information found so far

Initialize max\_profit\_info to (0, 0, 0, 0) // This will hold the best stock index, buy day, sell day, and max profit

# Iterate over each stock along with its daily prices

For each stock\_index and prices list in the matrix:

# Assume the minimum price is the first price of the stock

Set min\_price to the first price in prices

# Start with a maximum profit of zero since we haven't calculated profit yet

Initialize max\_profit to 0

# Assume the best day to buy is the first day, starting at 1 since we're using 1-indexing

Initialize buy\_day to 1

# Assume the best day to sell is the first day, same reasoning as buy\_day

Initialize sell\_day to 1

# Loop through each price for the current stock

For each current\_day and price in prices:

# Check if the current price is lower than the previously found minimum price

If price is less than min\_price:

# If a new minimum is found, update min\_price

Update min\_price to the current price

# Also, update the buy\_day since we found a cheaper price to buy at

Update buy\_day to current\_day + 1 // Convert from 0-indexed to 1-indexed

# Calculate the profit if we sold the stock on the current day

Calculate current\_profit as price minus min\_price

# Check if selling today is better than any previous sell day

If current\_profit is greater than max\_profit:

# If so, update max\_profit to the current\_profit

Update max\_profit to current\_profit

# Update the sell\_day to the current day

Update sell\_day to current\_day + 1 // Convert from 0-indexed to 1-indexed

# After processing all prices for the current stock,

# check if the profit from this stock is better than the profit from previous stocks

If max\_profit for the current stock is greater than the max\_profit in max\_profit\_info:

# If it is, update max\_profit\_info with the new best profit information

Update max\_profit\_info with the current stock\_index + 1, buy\_day, sell\_day, and max\_profit

# After going through all the stocks, return the information about the stock that gives the maximum profit

Return max\_profit\_info