Midterm – 200 points
Due April 15, 11:55PM, via NYU Brightspace

Note: There are 3 questions adding up to 300 points. 200 points is a perfect score.

# 1. Hadoop:

## Regional Temperature Analysis – 100 points

Dataset: temperature_data.txt

**Problem Statement**

You have a dataset containing temperature readings from weather stations across multiple countries. Each line in the dataset contains:

- Station ID (string)
- Country code (2-letter string)
- Region code (3-letter string)
- Timestamp (YYYY-MM-DD HH:MM)
- Temperature reading (float, in Celsius)
- Humidity (integer percentage)

Example data:

```
ST001,US,NYE,2024-01-15 14:30:21,22.4,65
ST045,CA,ONT,2024-01-15 14:32:15,-5.2,48
ST001,US,NYE,2024-01-15 15:30:21,23.1,64
```

SOLVE:

1.1 Calculate the temperature anomaly for each reading, defined as the difference between that reading and the average temperature for that station in its region.

1.2 Identify the top 3 regions with the highest temperature volatility, where volatility is measured as the standard deviation of temperature anomalies within each region.

**The Challenge**

1. You need to calculate regional averages before determining anomalies, but you can't hold all data in memory.
2. There's no global state in MapReduce streaming, so computing standard deviations requires thought.
3. Computing top-3 regions requires aggregation across the entire dataset.

**Requirements**

1. Implement this solution using MapReduce Streaming with any language of your choice (Python, Ruby, Perl, etc.)
2. Your solution must scale to handle datasets too large to fit in memory.
3. The solution can be solved using multiple MapReduce jobs that communicate through intermediate files, environment variables and/or command line arguments.
4. Explain the data flow between your MapReduce jobs and how each overcomes the stateless limitation.

# 2. Spark:

## E-Commerce Data Analysis – 100 points

Datasets: customers.csv, orders.csv, order_items.csv

### Problem Statement

Analyze customer purchasing patterns and identify high-value customers with specific behaviors.

### Solve:

#### 2.1 Customer Spending Analysis

Compute total amount spent by each customer. Note there are cancelled orders that should not count. Show which customers spend the most money overall

### 2.2 Category Preference Analysis

Compute the most popular product categories for each customer tier (Gold, Silver, Bronze)

### 2.3 Price Range Preferences

Show how different customer tiers distribute their spending across budget, mid-range, and premium products, i.e. whether Gold tier customers buy more premium products than Silver or Bronze customers

Use this functions:

```
def classify_price(price):
    if price < 50.0:
        return "Budget"
    elif price < 200.0:
        return "Mid-range"
    else:
        return "Premium"
```

### 2.4 Top Customer Identification

Compute the top 2 highest-spending customers within each tier.

Use this formula:

## 3. Spark EXTRA CREDIT:

## Customer E-Commerce Analysis – 100 points

Datasets: from Q2.

### 3.2 Customer Value Segmentation

Classify customers as "High Value," "Medium Value," or "Low Value" based on:

* Recency: How recently they've made a purchase

* Frequency: How often they make purchases

* Monetary value: Average order size

Use this function definition:

```
def segment_customer(recency_days, frequency, avg_value):
    # Score each dimension on a scale of 1-3
    recency_score = 3 if recency_days <= 30 else (2 if recency_days <= 90 else 1)
    frequency_score = 3 if frequency >= 3 else (2 if frequency >= 2 else 1)
    value_score = 3 if avg_value >= 200 else (2 if avg_value >= 100 else 1)

    # Calculate the overall score
    total_score = recency_score + frequency_score + value_score

    # Assign segment based on total score
    if total_score >= 8:
        return "High Value"
    elif total_score >= 5:
        return "Medium Value"
    else:
       return "Low Value"
```

### 3.3. Top 3 Categories

Compute the top 3 categories each customer purchases most frequently