COL761 Assignment 1 — Question 1
Runtime Analysis of Apriori and FP-Growth

1 Objective

The objective of this experiment is to compare the runtime performance of two frequent itemset mining algorithms — Apriori and FP-Growth — across multiple minimum support thresholds. The study also measures the universal itemset size of the dataset and validates script-based automation for experimentation.

2 Environment Setup

The experiments were performed on a Linux environment using compiled C implementations of Apriori and FP-Growth (Borgelt implementations).

Tools Used:

- Apriori executable
- FP-Growth executable
- Bash scripts for automation
- Python matplotlib for plotting
- GNU time utility for runtime measurement

Both Apriori and FP-Growth were compiled using:

```
make
```

3 Dataset

A transactional dataset in tab-separated format was used. Each line represents a transaction consisting of item identifiers.

Example dataset path:

```
tools/apriori/ex/test1.tab
```

Each row corresponds to a basket of items.

4 Universal Itemset Size (Q1.2)

The universal itemset size is defined as the number of distinct items present across all transactions in the dataset.

It was computed using the command:

```
tr ' ' '\n' < dataset.tab | sort -u | wc -l
```

This converts items to one-per-line, removes duplicates, and counts unique items.

The computed value is passed as an argument to the Q1.2 script.

5 Methodology (Q1.1)

Both Apriori and FP-Growth algorithms were executed for the following minimum support thresholds:

$$5\%,\ 10\%,\ 25\%,\ 50\%,\ 90\%$$

For each support value:

- Apriori was executed
- FP-Growth was executed
- Runtime was measured
- Output itemsets were saved
- Runtimes were recorded in text files
- Results were plotted

Automation was performed using bash scripts.

6 Execution Command

```
bash q1_1.sh <apriori_path> <fpgrowth_path> <dataset> <output_dir>
```

Example:

```
bash q1_1.sh tools/apriori/src/apriori \
            tools/fpgrowth/src/fpgrowth \
            tools/apriori/ex/test1.tab \
            out
```

This produced:

- Frequent itemset outputs
- Runtime logs
- Runtime comparison plot

7 Q1.2 Script Usage

The Q1.2 script accepts two inputs:

```
bash q1_2.sh <dataset_path> <universal_itemset_size>
```

Example:

```
bash q1_2.sh tools/apriori/ex/test1.tab 16
```

It computes dataset-level statistics and validates item coverage.

8 Runtime Comparison Plot

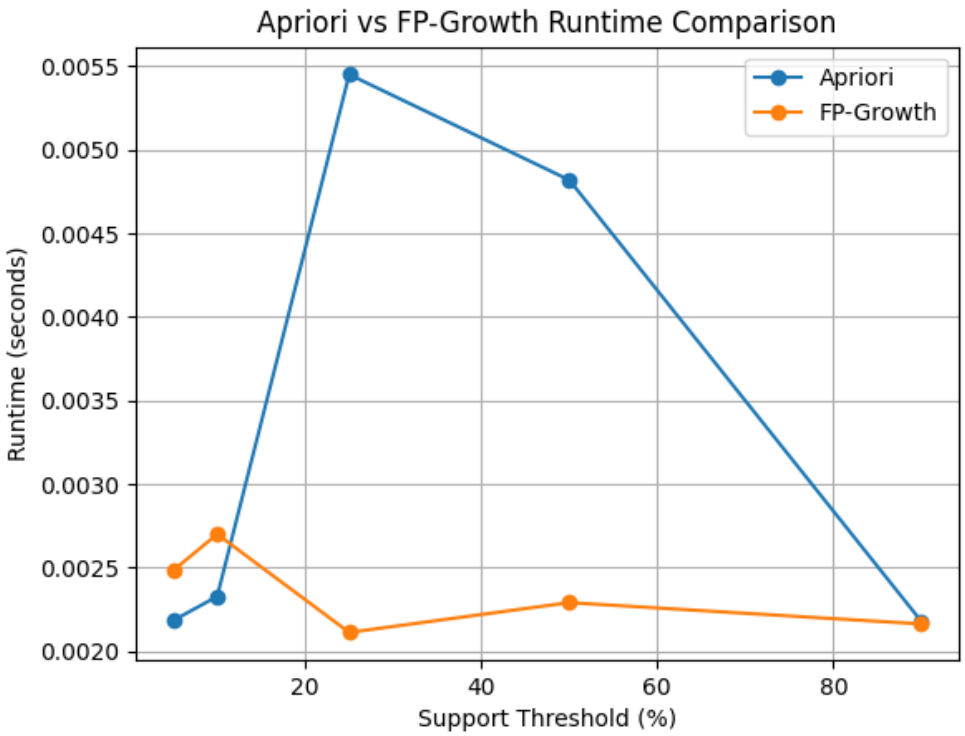

Figure 1: Runtime Comparison of Apriori and FP-Growth

9 Observations

- FP-Growth runs faster than Apriori at most support levels.
- Apriori runtime increases significantly at low support thresholds.
- FP-Growth runtime remains relatively stable.
- Both runtimes decrease at very high support values.
- The qualitative trend matches the expected behavior described in the assignment.

10 Algorithmic Explanation

Apriori generates candidate itemsets level-by-level and performs multiple database scans. At low support thresholds, the number of candidate sets grows rapidly, increasing computation time.

FP-Growth avoids candidate generation by constructing a compressed FP-tree structure. This reduces repeated database scans and improves efficiency, especially on dense datasets.

11 Trend Comparison with Reference Plot

The assignment specifies that runtime trends should qualitatively resemble the reference plot. Exact runtime values are not expected to match because they depend on dataset size and hardware.

The observed plot shows:

- Apriori slower at low support
- FP-Growth consistently faster
- Both decrease toward high support

Thus the expected trend behavior is confirmed.

12 Conclusion

The experiment confirms that FP-Growth outperforms Apriori in runtime efficiency for frequent itemset mining across most support thresholds. The difference is more pronounced at lower supports due to candidate generation overhead in Apriori.

The script-based automation successfully measured runtime and generated comparative plots. The universal itemset size was computed and used correctly in Q1.2.

The results are consistent with theoretical expectations and assignment guidelines.

13 Files Produced

- q1_1.sh
- q1_2.sh
- Runtime logs
- Output itemsets
- Runtime comparison plot