Course name: Data Science (ITE4005)

Professor: Dong-Kyu Chae (email: dongkyu@hanyang.ac.kr)

Teaching Assistant: Byeongtae Park (email: byeongtae@hanyang.ac.kr)

< Programming Assignment #1 >

10 Mar. 2022

Due Date: 28 March 2022, 11:59 pm

1. Environment

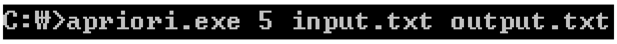
- OS: Windows, Mac OS, or Linux
- Languages: Java or Python (any version is ok)

2. Goal: find association rules using the **Apriori** algorithm

3. Requirements

The program must meet the following requirements:

- Execution file name: apriori.exe (or, apriori.py, apriori.etc…)
- Execute the program with three arguments: minimum support, input file name, output file name
  - ■ Example:



```
C:₩>apriori.exe 5 input.txt output.txt
```

  - Minimum support = 5%, input file name = 'input.txt', output file name = 'output.txt'

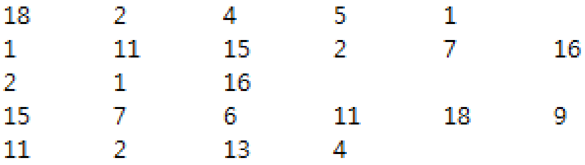
- Input file format (.txt)

  [*item_id*]\t[*item_id*]\n

  [*item_id*]\t[*item_id*]\t[*item_id*]\t[*item_id*]\t[*item_id*]\n

  [*item_id*]\t[*item_id*]\t[*item_id*]\t[*item_id*]\n

  - ■ Row: transaction
  - ■ *item_id* is a numerical value
  - ■ There is no duplication of items in each transaction
  - ■ Example:



```
18      2       4       5       1
1       11      15      2       7       16
2       1       16
15      7       6       11      18      9
11      2       13      4
```

- Output file format (.txt)

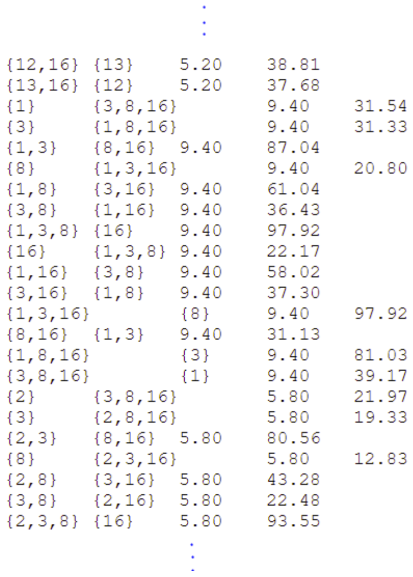
  [*item_set*]\t[*associative_item_set*]\t[*support*(%)]\t[*confidence*(%)]\n

  [*item_set*]\t[*associative_item_set*]\t[*support*(%)]\t[*confidence*(%)]\n

  - ■ [*item_set*]\t[*associative_item_set*]: association rules with minimum support

- [*item_set*] → [*associative_item_set*]
- Use braces to represent item sets: {[*item_id*],[*item_id*],…} (*Important!!*)
  - e.g., {0}, {0,4}, {0,3,1}

- *Support*: probability that a transaction contains [*item_set*] ∪ [*associative_item_set*]
- *Confidence*: conditional probability that a transaction having [*item_set*] also contains [*associative_item_set*]
- The order of output is unimportant.
- The value of support and confidence should be rounded to two decimal places.
  - e.g., 24.631 rounded to two decimal places should become 24.63.
- An additional penalty will be imposed if you don’t keep the output file format.
- Example:



```
                    ⋮
{12,16} {13}    5.20    38.81
{13,16} {12}    5.20    37.68
{1}     {3,8,16}        9.40    31.54
{3}     {1,8,16}        9.40    31.33
{1,3}   {8,16}  9.40    87.04
{8}     {1,3,16}        9.40    20.80
{1,8}   {3,16}  9.40    61.04
{3,8}   {1,16}  9.40    36.43
{1,3,8} {16}    9.40    97.92
{16}    {1,3,8} 9.40    22.17
{1,16}  {3,8}   9.40    58.02
{3,16}  {1,8}   9.40    37.30
{1,3,16}        {8}     9.40    97.92
{8,16}  {1,3}   9.40    31.13
{1,8,16}        {3}     9.40    81.03
{3,8,16}        {1}     9.40    39.17
{2}     {3,8,16}        5.80    21.97
{3}     {2,8,16}        5.80    19.33
{2,3}   {8,16}  5.80    80.56
{8}     {2,3,16}        5.80    12.83
{2,8}   {3,16}  5.80    43.28
{3,8}   {2,16}  5.80    22.48
{2,3,8} {16}    5.80    93.55
                   ⋮
```

- **Note: Please make sure to match the output format!**
  If the format is not correct, you can't get any score.

4. Submission

- Please submit the program files and the report **to GitLab**
  - Report
    - The file format of report must be *.pdf.
    - Guideline
      - ✓ Summary of your algorithm
      - ✓ Detailed description of your codes (for each function)
      - ✓ Instructions for compiling your source codes at TA's computer (e.g. screenshot) (*Important!!*)
      - ✓ Any other specification of your implementation and testing
  - Program files
    - A executable file (.exe or .py)
    - All source files
      - ✓ MakeFile if you use Linux
  - Note: submission details for GitLab will be announced later.

5. Penalty

- Late submission
  - 1 week delay: 20%
  - 2 weeks delay: 50%
  - Delay more than 2 weeks: 100%
- Requirements unsatisfied
  - Penalty up to 100% will be given depending on how the requirements are well-satisfied