COMP 9102 Data Management and Information Retrieval

Assignment 1

Implementation of query operators

Tianle WANG
Univ No. 3030096596
The University of Hong Kong

Date: September 27, 2022

Preparation

0.1 Environment Requirement

```
MacOS Monterey = 12.6 (OS that can run python3 command)
python = 3.8
pandas = 1.1.3 (just handle input)
```

0.2 Data

Put `R.tsv`, `R_sorted.tsv` and `S_sorted.tsv` in data directory that is at the same level as *.py files.

0.3 Results

create ouput directory that is at the same level as *.py files.

1 Merge-join

Run the following command in your terminal to execute the program:

```
python3 merge_join.py
```

1.1 Program

```
'''
1. lines should be read once only.
2. not read the entire files in data structures in memory. read each line and find
    results.
3. when join attribute of the next line is the same as previous line in file1, use
    buffer to record lines in file2. no need to re-read file2
'''
import csv
import datetime

fname1 = 'data/R_sorted.tsv'
fname2 = 'data/S_sorted.tsv'
output = 'output/RjoinS.tsv'

print(datetime.datetime.now())

f2 = open(fname2, 'r', encoding='utf-8')
line2 = f2.readline()
line2 = line2.strip('\n').split('\t')
with open(output, 'w', newline='') as f, open(fname1, 'r', encoding='utf-8') as f1:
    tsv_w = csv.writer(f, delimiter='\t')
    preLine1, preBuffer = [None] * 2
    maxLen = 0
    for line1 in f1:
        line1 = line1.strip('\n').split('\t')
        if preLine1 is not None and preLine1[0] == line1[0]: # use previous result
            for pb in preBuffer:
                tsv_w.writerow([line1[0], line1[1], pb[1]])
            continue
        buffer = []
        while line1[0] > line2[0]:
            line2 = f2.readline()
            line2 = line2.strip('\n').split('\t')
        while line2[0] == line1[0]:
            buffer.append(line2)
            tsv_w.writerow([line1[0], line1[1], line2[1]])
            line2 = f2.readline()
            line2 = line2.strip('\n').split('\t')
        maxLen = len(buffer) if maxLen < len(buffer) else maxLen # update maxLen
        preLine1 = line1
        preBuffer = buffer
f2.close()
print(maxLen)
```

```
print(datetime.datetime.now())
```

2 Union

Run the following command in your terminal to execute the program:

```
python3 union.py
```

2.1 Program

```
'''
1. lines should be read once only.
2. not read the entire files in data structures in memory.
3. no buffer.
4. eliminating duplicates.
'''
import csv
import datetime

fname1 = 'data/R_sorted.tsv'
fname2 = 'data/S_sorted.tsv'
output = 'output/RunionS.tsv'

print(datetime.datetime.now())

f1 = open(fname1, 'r', encoding='utf-8')
f2 = open(fname2, 'r', encoding='utf-8')
line1_str = f1.readline()
line2_str = f2.readline()
with open(output, 'w', newline='') as f:
    tsv_w = csv.writer(f, delimiter='\t')
    preline1, preline2 = [None] * 2
    while line1_str and line2_str:
        line1 = line1_str.strip('\n').split('\t')
        line2 = line2_str.strip('\n').split('\t')
        while line1_str and preline1 == line1: # continue the next line if duplicates
            line1_str = f1.readline()
            line1 = line1_str.strip('\n').split('\t')
        while line2_str and preline2 == line2: # continue the next line if duplicates
            line2_str = f2.readline()
            line2 = line2_str.strip('\n').split('\t')
```

```
        if line1[0] < line2[0] or (line1[0] == line2[0] and line1[1] < line2[1]): #
            line1 < line2
            tsv_w.writerow(line1)
            preline1 = line1
            line1_str = f1.readline()
            line1 = line1_str.strip('\n').split('\t')
        elif line1 == line2: # line1 = line2
            tsv_w.writerow(line1)
            preline1 = line1
            preline2 = line2
            line1_str = f1.readline()
            line1 = line1_str.strip('\n').split('\t')
            line2_str = f2.readline()
            line2 = line2_str.strip('\n').split('\t')
        else: # line1 > line2
            tsv_w.writerow(line2)
            preline2 = line2
            line2_str = f2.readline()
            line2 = line2_str.strip('\n').split('\t')

    if line1_str: # output line1 and duplicate
        while line1_str:
            if line1 == preline1:
                line1_str = f1.readline()
                line1 = line1_str.strip('\n').split('\t')
                continue
            tsv_w.writerow(line1)
            preline1 = line1
            line1_str = f1.readline()
            line1 = line1_str.strip('\n').split('\t')

    if line2_str: # output line2 and duplicate
        while line2_str:
            if line2 == preline2:
                line2_str = f2.readline()
                line2 = line2_str.strip('\n').split('\t')
                continue
            tsv_w.writerow(line2)
            preline2 = line2
            line2_str = f2.readline()
            line2 = line2_str.strip('\n').split('\t')

f1.close()
f2.close()
```

```
print(datetime.datetime.now())
```

3 Intersection

Run the following command in your terminal to execute the program:

```
python3 intersection.py
```

3.1 Program

```
'''
1. lines should be read once only.
2. not read the entire files in data structures in memory.
3. no buffer.
4. eliminating duplicates.
'''
import csv
import datetime

fname1 = 'data/R_sorted.tsv'
fname2 = 'data/S_sorted.tsv'
output = 'output/RintersectionS.tsv'

print(datetime.datetime.now())

f2 = open(fname2, 'r', encoding='utf-8')
line2 = f2.readline()
line2 = line2.strip('\n').split('\t')
with open(output, 'w', newline='') as f, open(fname1, 'r', encoding='utf-8') as f1:
    tsv_w = csv.writer(f, delimiter='\t')
    preLine1 = None
    for line1 in f1:
        line1 = line1.strip('\n').split('\t')
        if preLine1 == line1: # duplicates
            continue
        while line1 > line2:
            line2 = f2.readline()
            line2 = line2.strip('\n').split('\t')
        if line1 == line2: # intersect
            tsv_w.writerow(line1)
            line2 = f2.readline()
            line2 = line2.strip('\n').split('\t')
```

```
        preLine1 = line1
f2.close()
print(datetime.datetime.now())
```

4 Set difference

Run the following command in your terminal to execute the program:

```
python3 difference.py
```

4.1 Program

```
'''
1. lines should be read once only.
2. not read the entire files in data structures in memory.
3. no buffer.
4. eliminating duplicates.
'''
import csv
import datetime

fname1 = 'data/R_sorted.tsv'
fname2 = 'data/S_sorted.tsv'
output = 'output/RdifferenceS.tsv'

print(datetime.datetime.now())

f1 = open(fname1, 'r', encoding='utf-8')
line1_str = f1.readline()
line1 = line1_str.strip('\n').split('\t')
with open(output, 'w', newline='') as f, open(fname2, 'r', encoding='utf-8') as f2:
    tsv_w = csv.writer(f, delimiter='\t')
    preLine1, preLine2 = [None] * 2
    is_finished = False
    for line2 in f2:
        line2 = line2.strip('\n').split('\t')
        if preLine2 == line2: # duplicates
            continue
        while line1 <= line2:
            if line1 != line2:
                tsv_w.writerow(line1)
            preLine1 = line1
```

```
            line1_str = f1.readline()
            if not line1_str: # line1 is out
                is_finished = True
                break
            line1 = line1_str.strip('\n').split('\t')
            while line1 == preLine1:
                preLine1 = line1
                line1_str = f1.readline()
                line1 = line1_str.strip('\n').split('\t')

        if is_finished:
            break
        preLine2 = line2
f1.close()
print(datetime.datetime.now())
```

5 Grouping and Aggregation

Run the following command in your terminal to execute the program:

```
python3 groupby.py
```

5.1 Program

```
'''
1. read the entire files in data structures in memory.
2. sorted-merge
'''
import csv
import datetime
import pandas as pd

fname = 'data/R.tsv'
output = 'output/Rgroupby.tsv'

def MergeSort(lst):
    if len(lst) <= 1:
        return lst
    mid = int(len(lst)/2)
    left = MergeSort(lst[:mid])
    right = MergeSort(lst[mid:])
    return Merge(left, right)
```

```
def Merge(left, right):
    l, r = 0, 0
    result = []
    while l < len(left) and r < len(right):
        if left[l][0] < right[r][0]: # left < right
            result.append(left[l])
            l += 1
        elif left[l][0] == right[r][0]:
            tmp = left[l]
            tmp[1] += right[r][1]
            result.append(tmp)
            l += 1
            r += 1
        elif left[l][0] > right[r][0]:
            result.append(right[r])
            r += 1
    result += list(left[l:])
    result += list(right[r:])
    return result

print(datetime.datetime.now())

with open(output, 'w', newline='') as f:
    tsv_w = csv.writer(f, delimiter='\t')
    R = pd.read_csv(fname, sep='\t', header=None, names=['key', 'integer'],
        keep_default_na=False)
    R_dict = zip(R['key'], R['integer'])
    R_list = list(R_dict)
    R_list1 = []
    for elem in R_list:
        elem = list(elem)
        R_list1.append(elem)
    res = MergeSort(R_list1)
    tsv_w.writerows(res)
print(datetime.datetime.now())
```