AI융합 연계전공(010982)-인공지능의 이해 3강
Data Engineering for Artificial Intelligence

Jaewook Byun

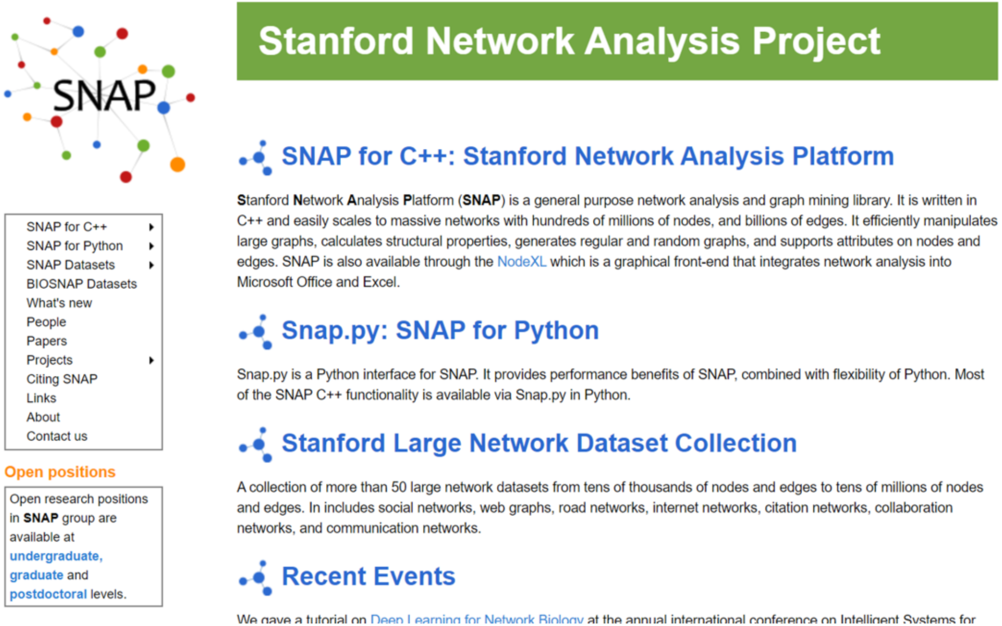
Ph.D., Assistant Professor, Department of Software, Sejong University

Table of Contents

1. Introduction to Data Engineering for Artificial Intelligence (Big Data Processing)
2. Data Collection & Processing

Open Dataset

- SNAP (Stanford Network Analysis Project)
  - A collection of more than 50 large network datasets from tens of thousands of nodes and edges to tens of millions of nodes and edges. In includes social networks, web graphs, road networks, internet networks, citation networks, collaboration networks, and communication networks.
  - https://snap.stanford.edu/data/



Open positions

Open research positions in **SNAP** group are available at **undergraduate**, **graduate** and **postdoctoral** levels.

Stanford Network Analysis Project

SNAP for C++: Stanford Network Analysis Platform

Stanford **N**etwork **A**nalysis **P**latform (**SNAP**) is a general purpose network analysis and graph mining library. It is written in C++ and easily scales to massive networks with hundreds of millions of nodes, and billions of edges. It efficiently manipulates large graphs, calculates structural properties, generates regular and random graphs, and supports attributes on nodes and edges. SNAP is also available through the NodeXL which is a graphical front-end that integrates network analysis into Microsoft Office and Excel.

Snap.py: SNAP for Python

Snap.py is a Python interface for SNAP. It provides performance benefits of SNAP, combined with flexibility of Python. Most of the SNAP C++ functionality is available via Snap.py in Python.

Stanford Large Network Dataset Collection

A collection of more than 50 large network datasets from tens of thousands of nodes and edges to tens of millions of nodes and edges. In includes social networks, web graphs, road networks, internet networks, citation networks, collaboration networks, and communication networks.

Recent Events

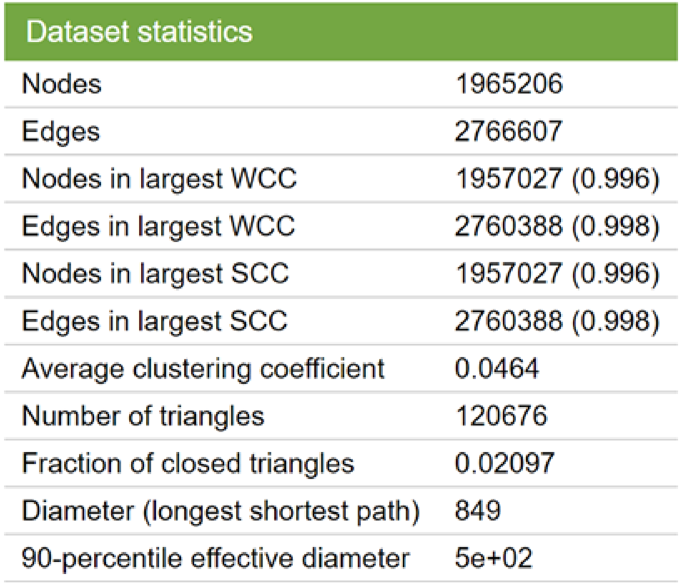
We gave a tutorial on Deep Learning for Network Biology at the annual international conference on Intelligent Systems for

A dataset used in the lecture

- California Road Network
  - Description: https://snap.stanford.edu/data/roadNet-CA.html
  - Download Link: https://snap.stanford.edu/data/roadNet-CA.txt.gz

D:\\data.txt

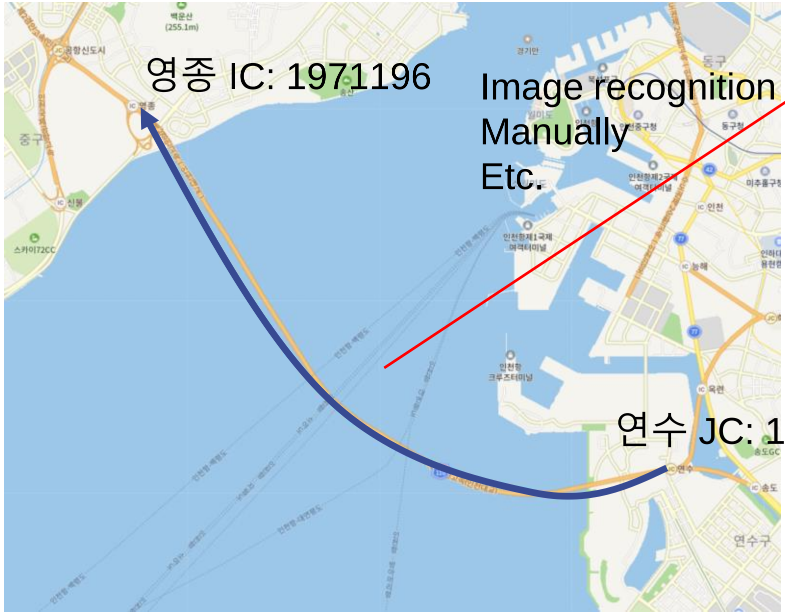
  - A road network of California. Intersections and endpoints are represented by nodes and the roads connecting these intersections or road endpoints are represented by undirected edges.



Dataset statistics	
Nodes	1965206
Edges	2766607
Nodes in largest WCC	1957027 (0.996)
Edges in largest WCC	2760388 (0.998)
Nodes in largest SCC	1957027 (0.996)
Edges in largest SCC	2760388 (0.998)
Average clustering coefficient	0.0464
Number of triangles	120676
Fraction of closed triangles	0.02097
Diameter (longest shortest path)	849
90-percentile effective diameter	5e+02

A dataset used in the lecture

- California Road Network
  - A road network of California. Intersections and endpoints are represented by nodes and the roads connecting these intersections or road endpoints are represented by undirected edges.



영종 IC: 1971196

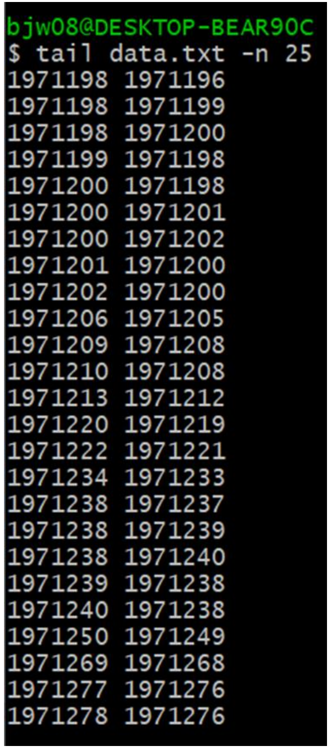
Image recognition
Manually
Etc.

연수 JC: 1971198

1971198→1971196

Open

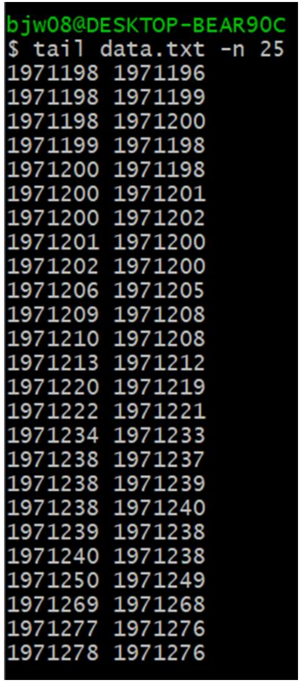
연수 JC -> 영종 IC ?



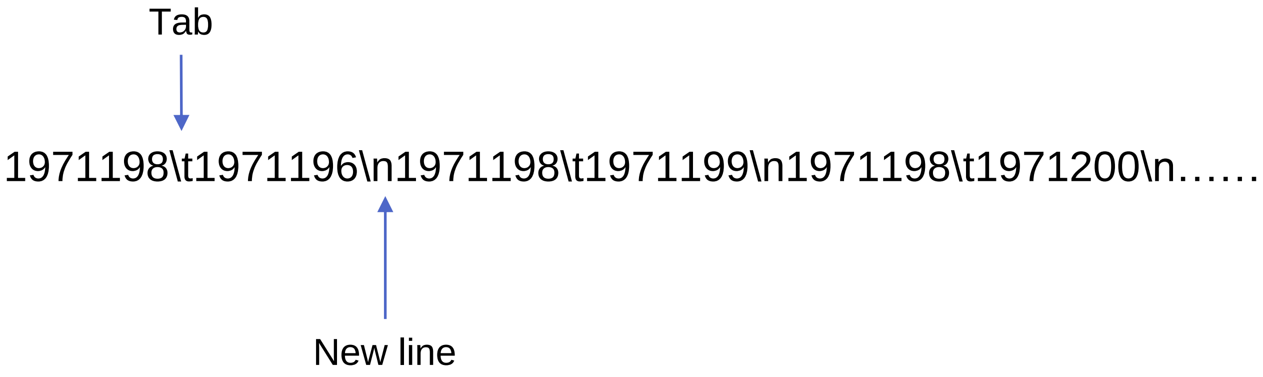
```
bjw08@DESKTOP-BEAR90C
$ tail data.txt -n 25
1971198 1971196
1971198 1971199
1971198 1971200
1971199 1971198
1971200 1971198
1971200 1971201
1971200 1971202
1971201 1971200
1971202 1971200
1971206 1971205
1971209 1971208
1971210 1971208
1971213 1971212
1971220 1971219
1971222 1971221
1971234 1971233
1971238 1971237
1971238 1971239
1971238 1971240
1971239 1971238
1971240 1971238
1971250 1971249
1971269 1971268
1971277 1971276
1971278 1971276
```

A dataset used in the lecture

- California Road Network
  - data.txt



```
bjw08@DESKTOP-BEAR90C
$ tail data.txt -n 25
1971198 1971196
1971198 1971199
1971198 1971200
1971199 1971198
1971200 1971198
1971200 1971201
1971200 1971202
1971201 1971200
1971202 1971200
1971206 1971205
1971209 1971208
1971210 1971208
1971213 1971212
1971220 1971219
1971222 1971221
1971234 1971233
1971238 1971237
1971238 1971239
1971238 1971240
1971239 1971238
1971240 1971238
1971250 1971249
1971269 1971268
1971277 1971276
1971278 1971276
```



Tab

1971198\t1971196\n1971198\t1971199\n1971198\t1971200\n……

New line

Problem 1: # of roads

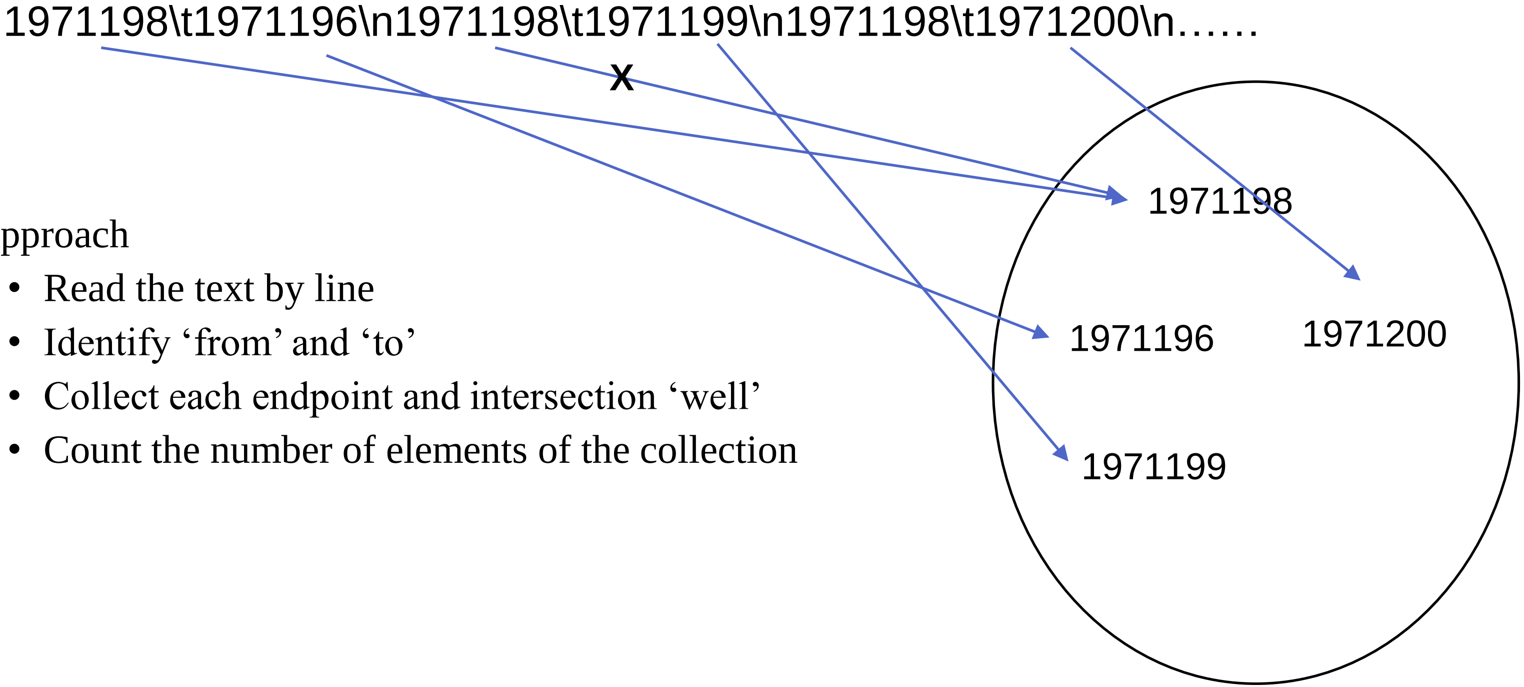
- California Road Network
  - How many roads are described in the dataset?

```
1971198\t1971196\n1971198\t1971199\n1971198\t1971200\n……
```

- Approach
  - Read the text by line
  - Count the number of lines

Problem 2: # of endpoints and intersection

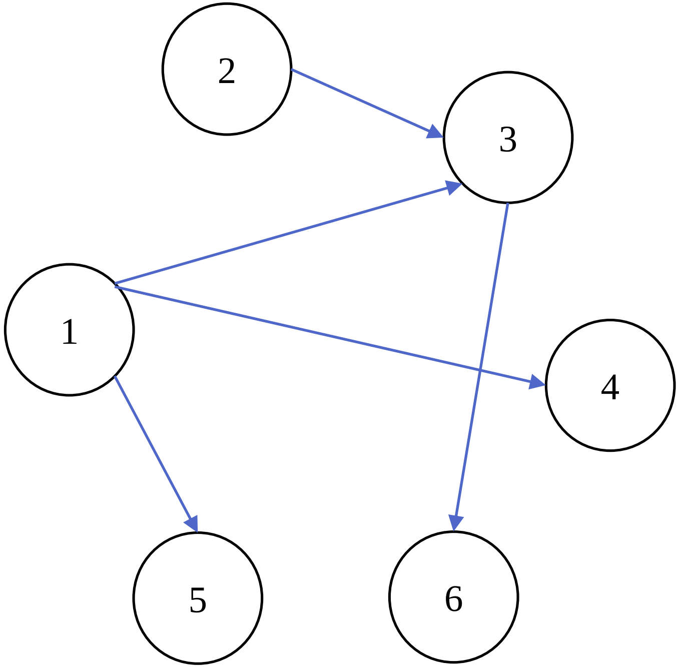
- California Road Network
  - How many endpoints and intersection are described in the dataset?



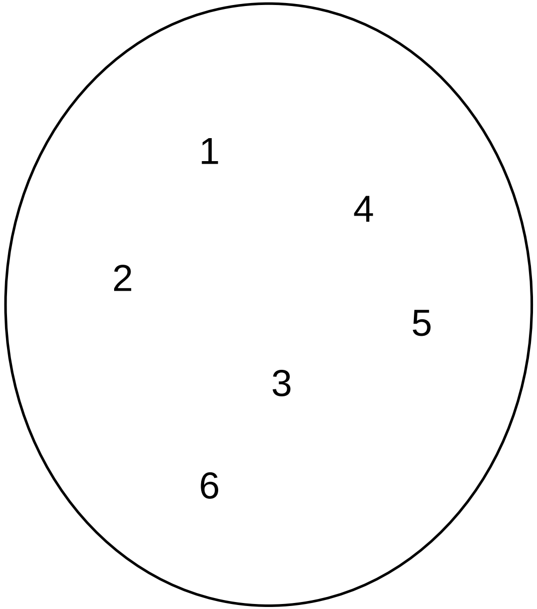
- Approach
  - Read the text by line
  - Identify ‘from’ and ‘to’
  - Collect each endpoint and intersection ‘well’
  - Count the number of elements of the collection

Problem 2: # of endpoints and intersection

- California Road Network
  - Using a sample dataset
    - 1 -> 3
    - 1 -> 4
    - 1 -> 5
    - 2 -> 3
    - 3 -> 6



A collection of
Non-redundant endpoints and
intersection

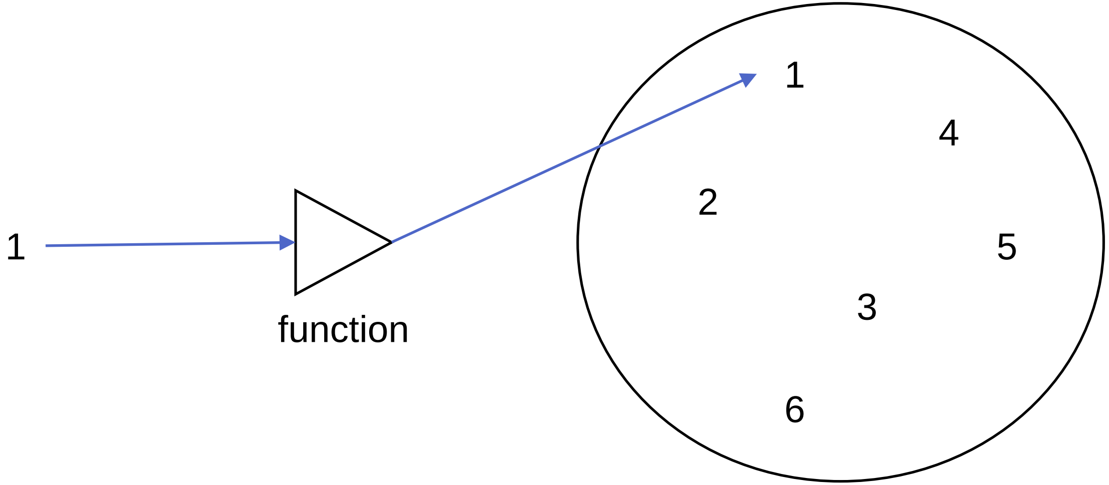


Problem 2: # of endpoints and intersection

- California Road Network
  - Using a sample dataset
    - 1, 3, 4, 5, 2, 3, 6

DS1	1	3	4	5	2	6

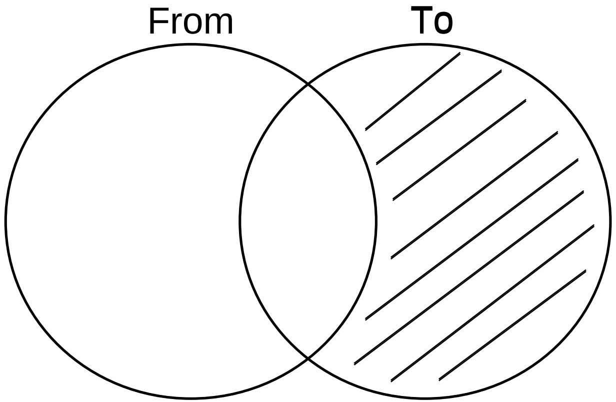
DS2



Problem 3: # of deadends

- California Road Network
    - # of dead ends

    - Think it again in a programmer's way
        - Dead end:
            - Only in a right part
            - == In a right part but not in a left part
            - == To – From (Mathematically)

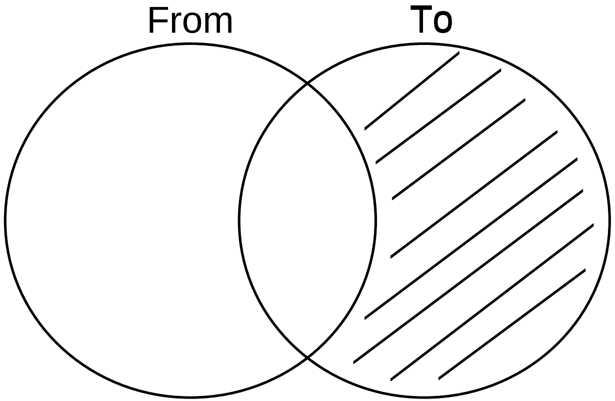


- Approach
    - Collect 'from‘ and ‘to' separately
    - See each from element
    - If the element is in ‘to'
        - Remove the element from ‘to'
    - Count of the size of 'to'

Problem 4: # of roads

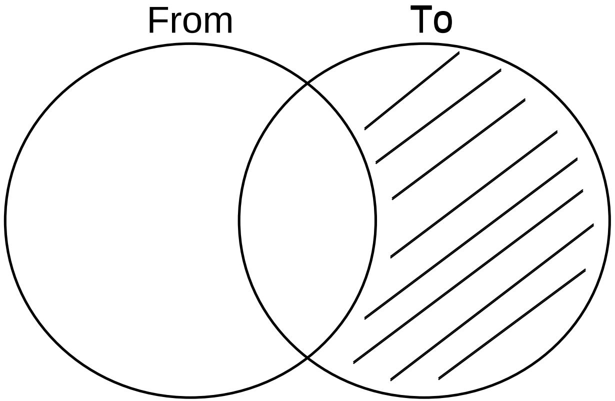
- California Road Network
  - # of roads

  - Is it possible to compute # of roads from
    - ‘from‘ set and ‘to’ set?



Problem 4: # of roads

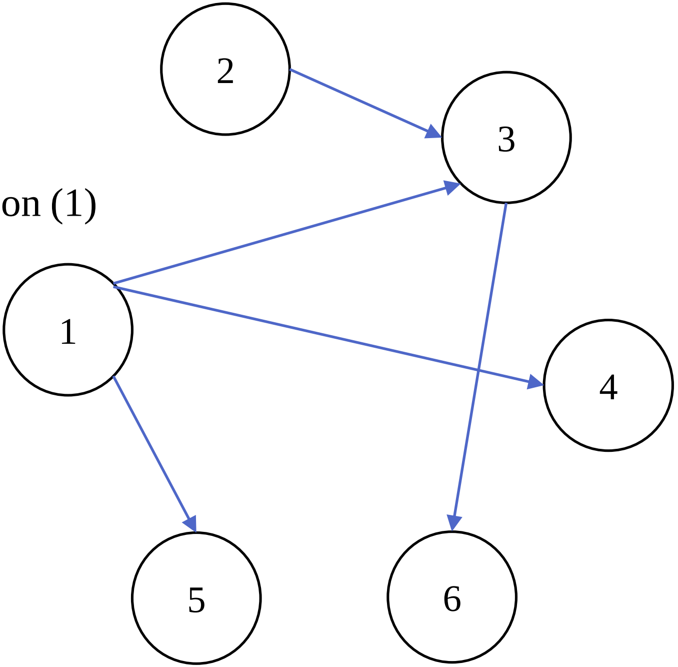
- California Road Network
  - # of roads
  - Is it possible to compute # of roads from
    - ‘from‘ set and ‘to’ set?
    - → No, because some information disappear



Problem 5: Reachable endpoints from a single source

- California Road Network
    - The reachable endpoints from a single source

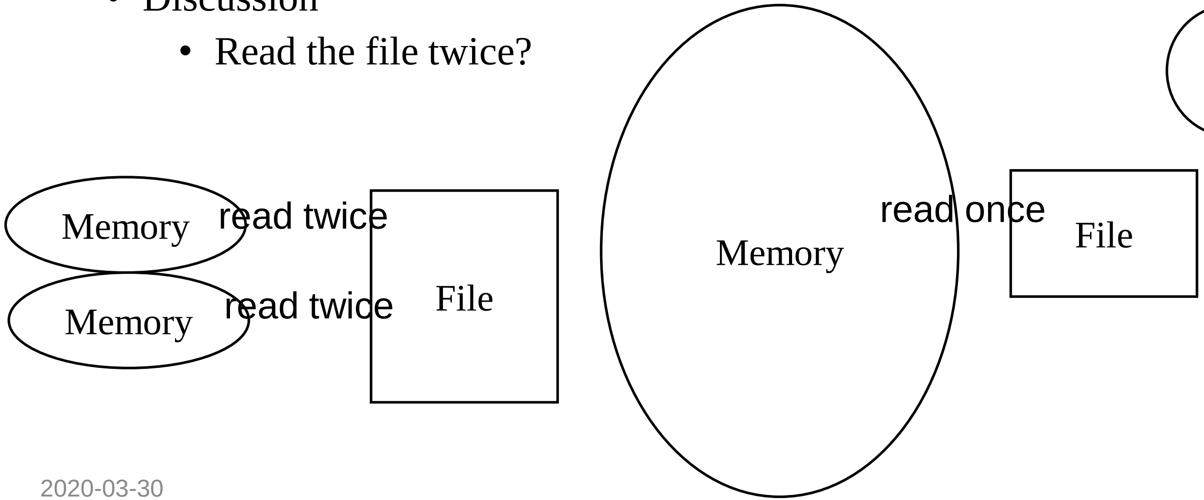
    - From '1‘ with 1 step?
        - 3, 4, 5
        - Approach
            - When a source is found in 'from', keep ‘to’ in a collection (1)

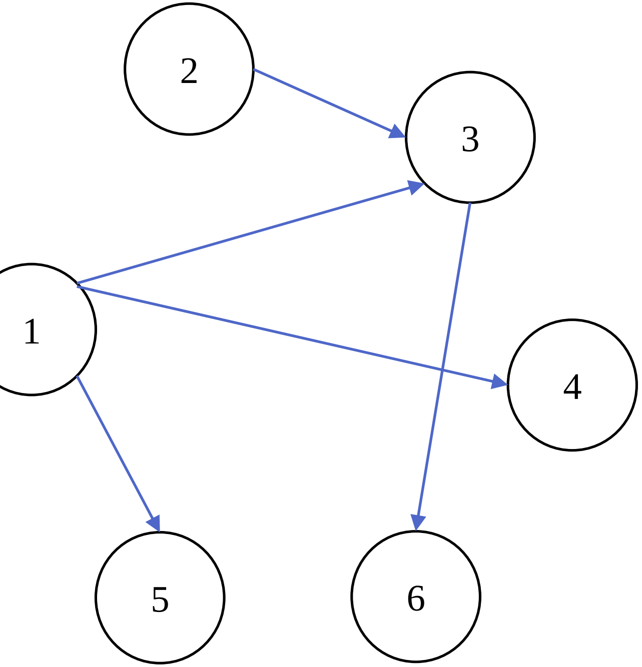


Problem 6: Reachable endpoints from a single source

- California Road Network
  - The reachable endpoints from a single source

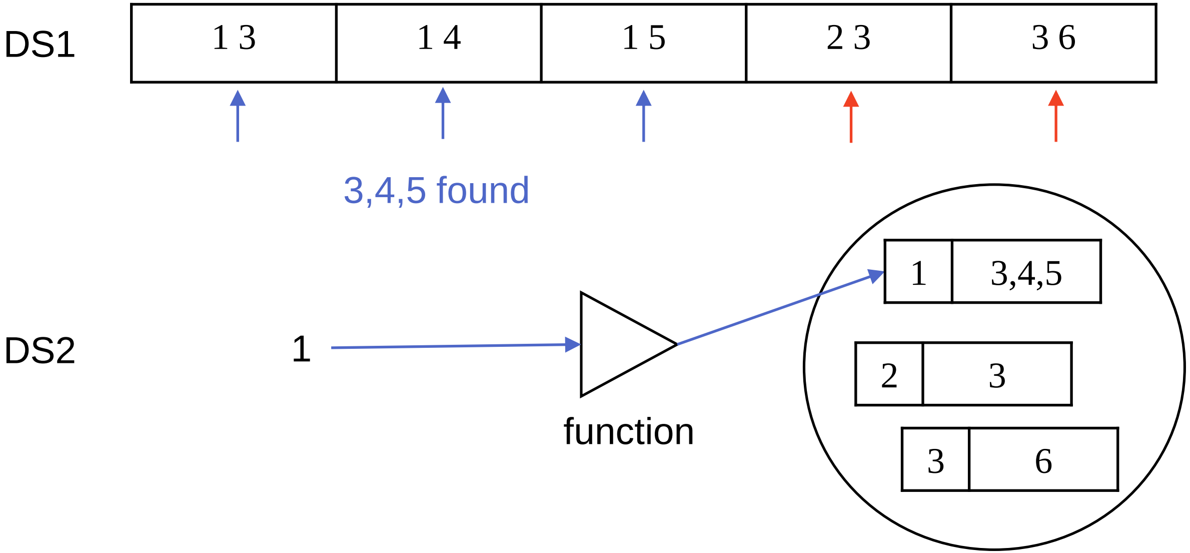
  - From ‘1’ with 1~2 steps?
    - 3,4,5,6
    - Approach
      - When (1) is found in 'from', keep ‘to’ in another col. (2)
  - Discussion
    - Read the file twice?





Problem 7: Reachable endpoints from a single source

- California Road Network
  - The reachable endpoints from a single source
  - From ‘1’ with 1~2 steps?
  - Discussion
    - More appropriate data structure?



DS1

1 3	1 4	1 5	2 3	3 6

3,4,5 found

DS2

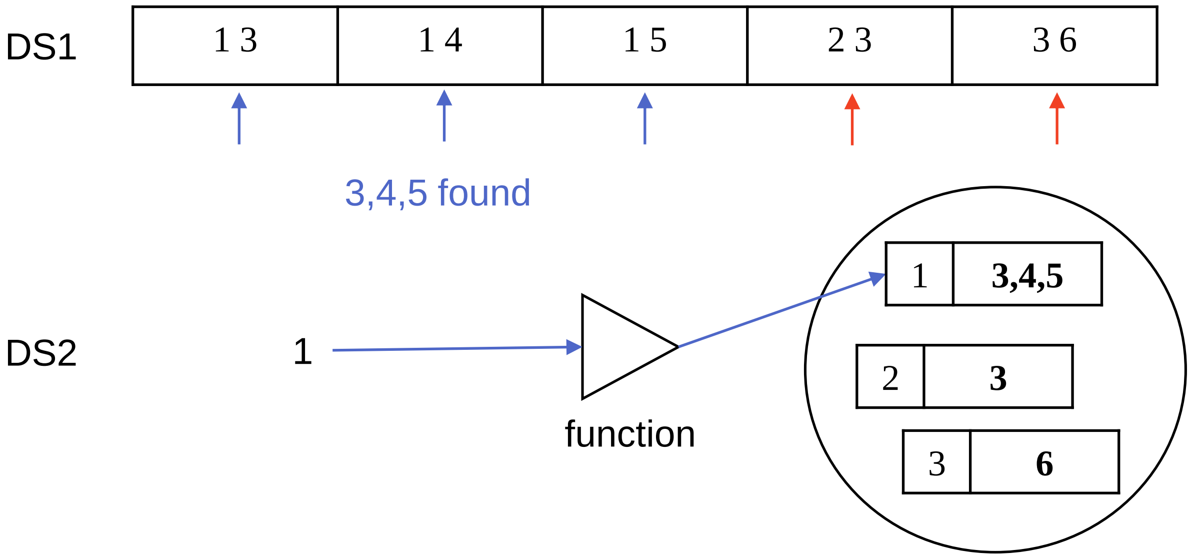
1

function

1	3,4,5
2	3
3	6

Problem 7: Reachable endpoints from a single source

- California Road Network
  - The reachable endpoints from a single source
  - From ‘1’ with 1~2 steps?
  - Discussion
    - More appropriate data structure? Works for the reversed way? No



DS1

1 3	1 4	1 5	2 3	3 6

3,4,5 found

DS2

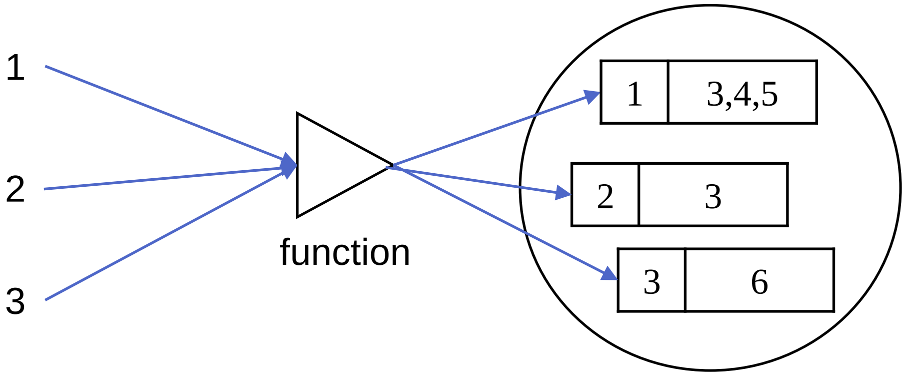
1	**3,4,5**
2	**3**
3	**6**

Problem 8: Reachable endpoints from a single source

- California Road Network
  - The reachable endpoints from a single source

  - From ‘1‘ with * step(s)?
    - Approach
      - Union (1),(2),…(n) until that is identical to a union of (1)~(n+1)

  - Discussion
    - Parallelism?
    - Always work?

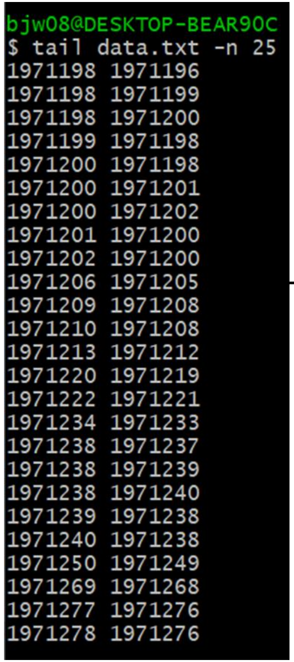


1	3,4,5
2	3
3	6

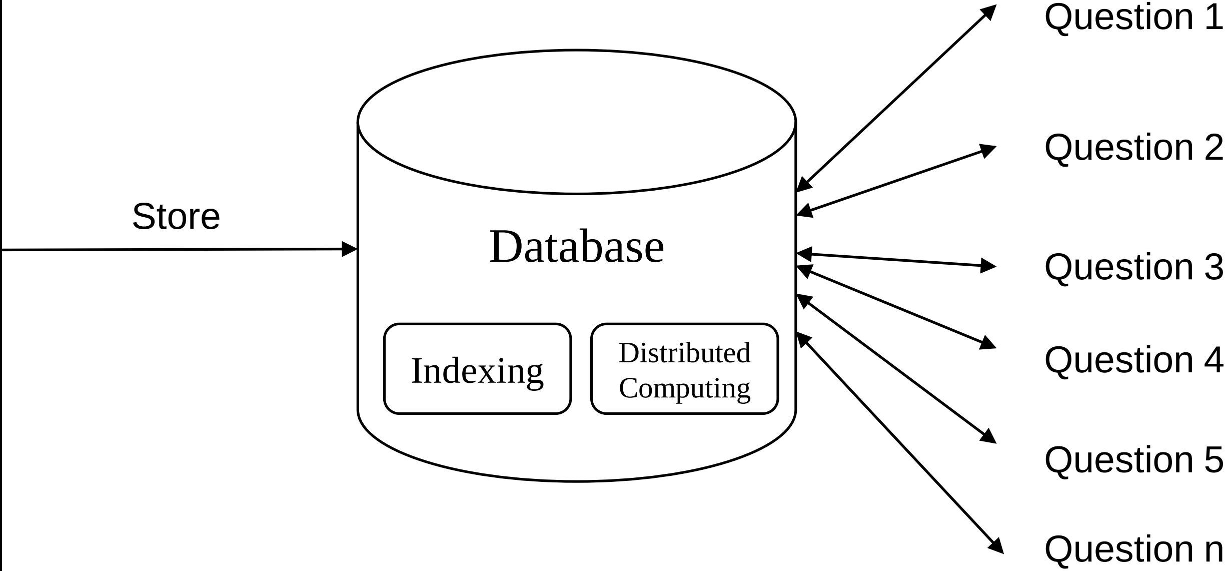
3 machines → X3 speed up?

Summary

- Data Engineering is your job



```
bjw08@DESKTOP-BEAR90C
$ tail data.txt -n 25
1971198 1971196
1971198 1971199
1971198 1971200
1971199 1971198
1971200 1971198
1971200 1971201
1971200 1971202
1971201 1971200
1971202 1971200
1971206 1971205
1971209 1971208
1971210 1971208
1971213 1971212
1971220 1971219
1971222 1971221
1971234 1971233
1971238 1971237
1971238 1971239
1971238 1971240
1971239 1971238
1971240 1971238
1971250 1971249
1971269 1971268
1971277 1971276
1971278 1971276
```



Q & A

Thank you for listening