

Spark Streaming

Spark - Exercises

1

Exercise #52

- GraphFrame
- Input:
 - The textual file vertexes.csv
 - It contains the vertexes of a graph
 - Each vertex is characterized by
 - id (string): user identifier
 - name (string): user name
 - age (integer): user age

2

Exercise #52

- The textual file edges.csv
 - It contains the edges of a graph
- Each edge is characterized by
 - src (string): source vertex
 - dst (string): destination vertex
 - linktype (string): "follow" or "friend"

3

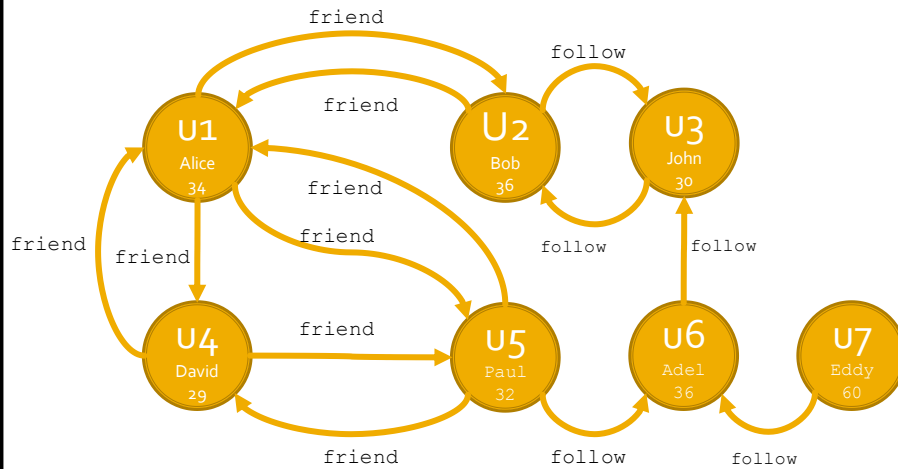
Exercise #52

- Output:
 - For each user with at least one follower, store in the output folder the number of followers
 - One user per line
 - Format: user id, number of followers
 - Use the CSV format to store the result

4

Exercise #52

Input graph example



5

Exercise #52

Result

id	numFollowers
U3	2
U6	2
U2	1

6

Exercise #53

- GraphFrame
- Input:
 - The textual file vertexes.csv
 - It contains the vertexes of a graph
 - Each vertex is characterized by
 - id (string): user identifier
 - name (string): user name
 - age (integer): user age

7

Exercise #53

- The textual file edges.csv
 - It contains the edges of a graph
- Each edge is characterized by
 - src (string): source vertex
 - dst (string): destination vertex
 - linktype (string): "follow" or "friend"

8

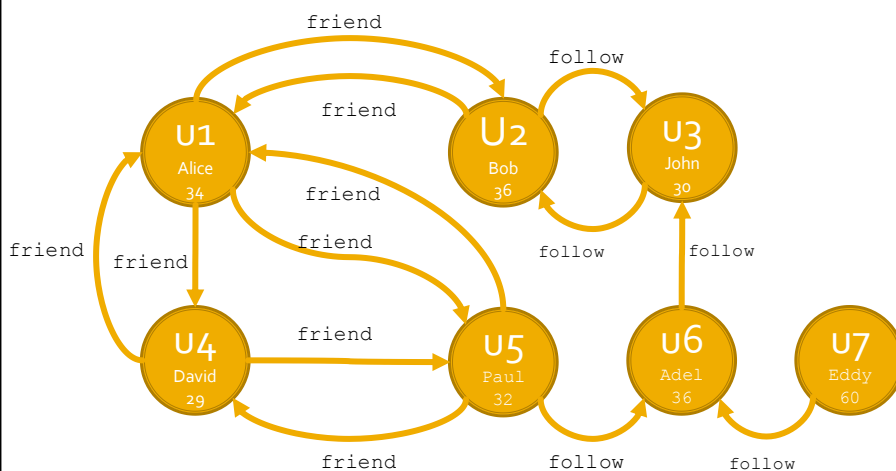
Exercise #53

- Output:
 - Consider only the users with at least one follower
 - Store in the output folder the user(s) with the maximum number of followers
 - One user per line
 - Format: user id, number of followers
 - Use the CSV format to store the result

9

Exercise #53

Input graph example



10

Exercise #53

- Result

id	numFollowers
u3	2
u6	2

11

Exercise #54

- GraphFrame
- Input:
 - The textual file vertexes.csv
 - It contains the vertexes of a graph
 - Each vertex is characterized by
 - id (string): user identifier
 - name (string): user name
 - age (integer): user age

12

Exercise #54

- The textual file edges.csv
 - It contains the edges of a graph
- Each edge is characterized by
 - src (string): source vertex
 - dst (string): destination vertex
 - linktype (string): "follow" or "friend"

13

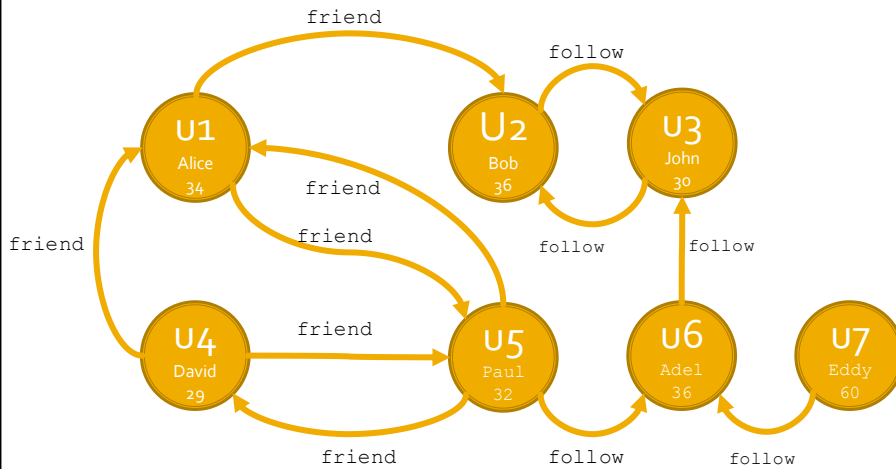
Exercise #54

- Output:
 - The pairs of users U_x, U_y such that
 - U_x is friend of U_y (link "friend" from U_x to U_y)
 - U_y is not friend of U_x (no link "friend" from U_y to U_x)
 - One pair U_x, U_y per line
 - Format: id U_x , id U_y
 - Use the CSV format to store the result

14

Exercise #54

Input graph example



15

Exercise #54

Result

IdFriend	IdNotFriend
U4	U1
U1	U2

16

Exercise #55

- GraphFrame
- Input:
 - The textual file vertexes.csv
 - It contains the vertexes of a graph
 - Each vertex is characterized by
 - id (string): vertex identifier
 - entityType (string): "user" or "topic"
 - name (string): name of the entity

17

Exercise #55

- The textual file edges.csv
 - It contains the edges of a graph
- Each edge is characterized by
 - src (string): source vertex
 - dst (string): destination vertex
 - linktype (string): "expertOf" or "follow" or "correlated"

18

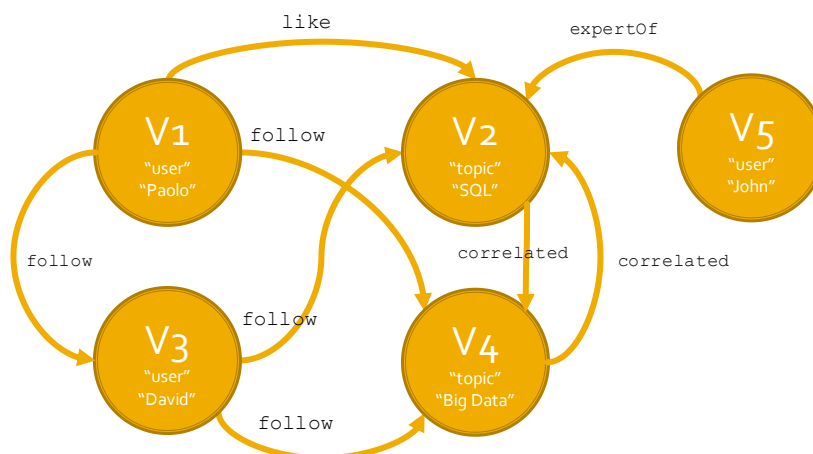
Exercise #55

- Output:
 - The followed topics for each user
 - One pair (user name, followed topic) per line
 - Format: username, followed topic
 - Use the CSV format to store the result

19

Exercise #55

Input graph example



20

Exercise #55

- Result

username	topic
Paolo	Big Data
David	SQL
David	Big Data

21

Exercise #56

- GraphFrame
- Input:
 - The textual file vertexes.csv
 - It contains the vertexes of a graph
 - Each vertex is characterized by
 - id (string): vertex identifier
 - entityType (string): "user" or "topic"
 - name (string): name of the entity

22

Exercise #56

- The textual file edges.csv
 - It contains the edges of a graph
- Each edge is characterized by
 - src (string): source vertex
 - dst (string): destination vertex
 - linktype (string): "expertOf" or "follow" or "correlated"

23

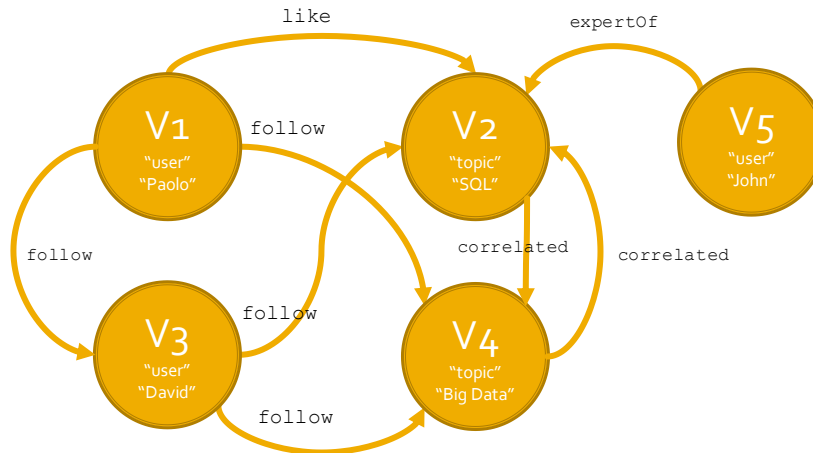
Exercise #56

- Output:
 - The names of the users who follow a topic correlated with the "Big Data" topic
 - One user name per line
 - Format: username
 - Use the CSV format to store the result

24

Exercise #56

Input graph example



25

Exercise #56

- Result

username
David

26

Exercise #57

- GraphFrame
- Input:
 - The textual file vertexes.csv
 - It contains the vertexes of a graph
 - Each vertex is characterized by
 - id (string): user identifier
 - name (string): user name
 - age (integer): user age

27

Exercise #57

- The textual file edges.csv
 - It contains the edges of a graph
- Each edge is characterized by
 - src (string): source vertex
 - dst (string): destination vertex
 - linktype (string): "follow" or "friend"

28

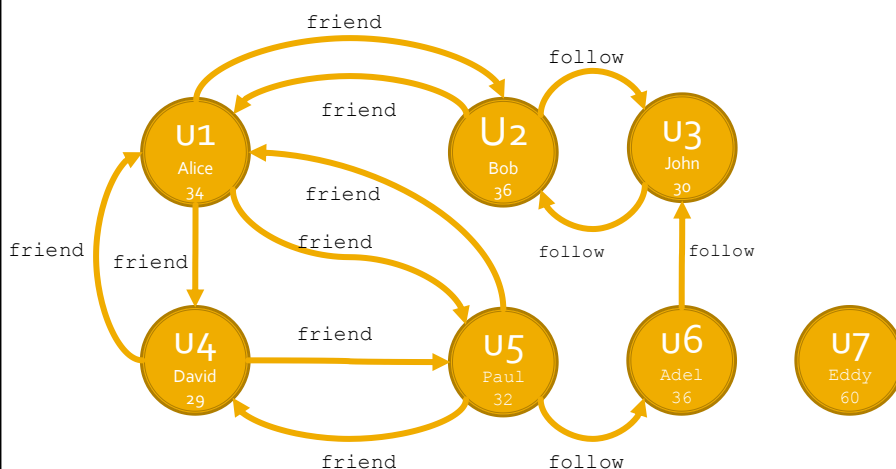
Exercise #57

- Output:
 - Select the users who can reach user u_1 in less than 3 hops (i.e., at most two edges)
 - Do not consider u_1 itself
 - For each of the selected users, store in the output folder his/her name and the minimum number of hops to reach user u_1
 - One user per line
 - Format: user name, #hops to user u_1
 - Use the CSV format to store the result

29

Exercise #57

Input graph example



30

Exercise #57

- Result

name	numHops
Bob	1
John	2
David	1
Paul	1