МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний аерокосмічний університет ім. М.Є. Жуковського

“Харківський авіаційний інститут”

Кафедра комп’ютерних систем, мереж і кібербезпеки

Лабораторна робота № 3

з дисципліни “ Великі дані для кібербезпеки”

Варіант № \_21\_

|  |  |  |
| --- | --- | --- |
| Виконав студент | гр. 555M | Орлов С.В. |
|  | (№ групи) | (П.І.Б.) |
| (підпис, дата) |  |  |
| Перевірив | д.т.н., професор, професор каф. 503 | |
|  | (науковий ступінь, вчене звання, посада) | |
|  | Фесенко Г. В. | |
| (підпис, дата) | (П.І.Б.) | |

2023

**Тема роботи**

Робота з графами з використанням Spark GraphX.

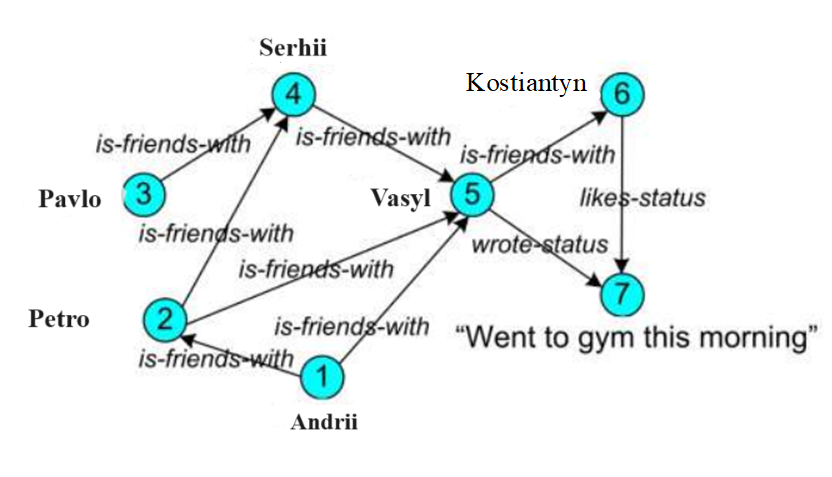
**Мета роботи**

Отримати навички роботи з графами з використанням Spark GraphX.

**Завдання**

Варіант 21

Перед виконанням роботи було завантажено та встановлено Spark на ОС Linux.



* 1. Загальні відомості про Spark GraphX

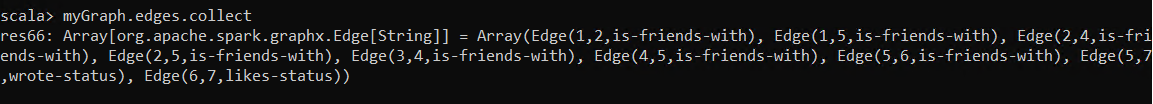
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Vertices RDD   |  |  | | --- | --- | | 1 | Andrii | | 2 | Petro | | 3 | Pavlo | | 4 | Serhii | | 5 | Vasyl | | 6 | Kostiantyn | | 7 | Went to gym this morning | | Edges RDD   |  |  |  | | --- | --- | --- | | Src | Dest | Attr | | 1 | 2 | is friends with | | 1 | 5 | is friends with | | 2 | 4 | is friends with | | 2 | 5 | is friends with | | 3 | 4 | is friends with | | 4 | 5 | is friends with | | 5 | 6 | is friends with | | 5 | 7 | wrote status | | 6 | 7 | likes status | |

1.2 Створення Graph і прості перетворення

1.2.1 Використаємо GraphX, щоб дістати з графу, представленого на рис. 2, тільки вершини:

|  |
| --- |
|  |

1.2.2 Дістанемо ребра з щойно отриманого графа



1.2.3 Ви також можете використовувати метод triplets() для об’єднання вершин і ребер на основі VertexId:

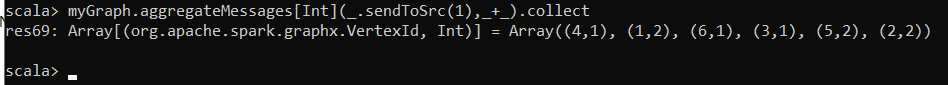
|  |
| --- |
|  |

1.2.4 До побудованому графу давайте додамо анотацію до кожного ребра «is–friends–with», коли у людини на початку ланцюжка дружби є ім’я, яке містить літеру a. Як ми додамо цю анотацію? Перетворюючи тип Edge з String в кортеж (String, Boolean), як показано в наступному лістингу. Клас EdgeTriplet знадобиться тут, оскільки нам потрібен доступ як до атрибуту ребра, так і до атрибуту вихідної вершини.

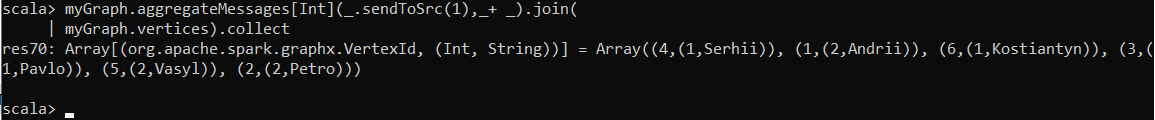
Зображення, що містить знімок екрана, текст, Шрифт, інформація

Автоматично згенерований опис

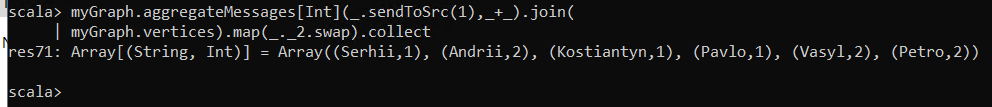
1.2.5 Визначимо кількість ребер, що виходять з кожної вершини, використовуючи метод aggregateMessages():



1.2.6 Покажемо для кожного номера вершини її назву і кількість ребер, що з неї виходять, використовуючи функцію join():



1.2.7 Приберемо номер вершини, залишивши тільки її ім’я і кількість ребер, що з неї виходять, використовуючи методи map() і swap():

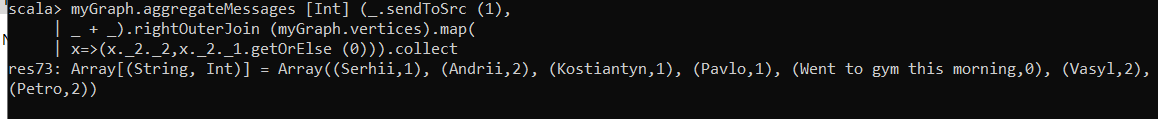


1.2.8 Щоб показати всі назви вершин (навіть ті, з яких не виходить жодного ребра), використовуємо rightOuterJoin ():

Зображення, що містить знімок екрана, текст, Шрифт, чорний

Автоматично згенерований опис

1.2.9 Почистимо попередній лістинг, позбувшись Some і None. Зробити це нам допоможе метод getOrElse():



1.3 Ітераційний Map / Reduce

Зображення, що містить текст, знімок екрана, Шрифт, ряд

Автоматично згенерований опис

Зображення, що містить текст, Шрифт, чорний, знімок екрана

Автоматично згенерований опис

Зображення, що містить текст, знімок екрана, Шрифт, програмне забезпечення

Автоматично згенерований опис

Зображення, що містить текст, знімок екрана, Шрифт

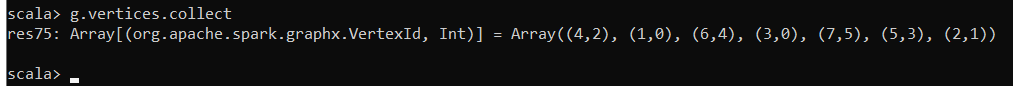
Автоматично згенерований опис

Вершина № 7 має найдовшу відстань до початкової вершини, яка становить 5.

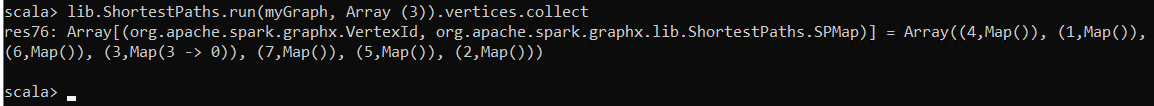
1.4 Використання Pregel API для ітераційних алгоритмів

Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис



1.5 Знаходження найменшої кількості переходів: алгоритм ShortestPaths



1.6 Визначення найкоротшого шляху на графі з позначеними вагами ребер

Зображення, що містить коло, ряд, схема, знімок екрана

Автоматично згенерований опис

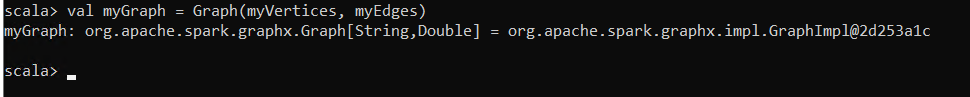
|  |  |
| --- | --- |
| Вага pебра | Варіант 21 |
| L1–2 | 7 |
| L1–3 | 7 |
| L1–4 | 9 |
| L2–4 | 9 |
| L2–6 | 7 |
| L3–4 | 9 |
| L3–5 | 7 |
| L4–5 | 3 |
| L4–7 | 7 |
| L4–8 | 4 |
| L5–8 | 8 |
| L6–9 | 8 |
| L6–7 | 3 |
| L7–9 | 8 |
| L7–10 | 7 |
| L8–10 | 7 |
| L9–10 | 7 |

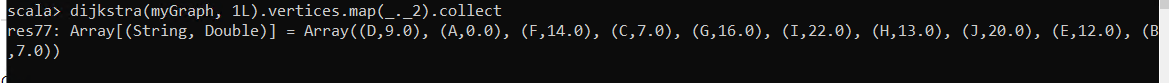
Зображення, що містить текст, знімок екрана, Мультимедійне програмне забезпечення, програмне забезпечення

Автоматично згенерований опис

Зображення, що містить текст, знімок екрана

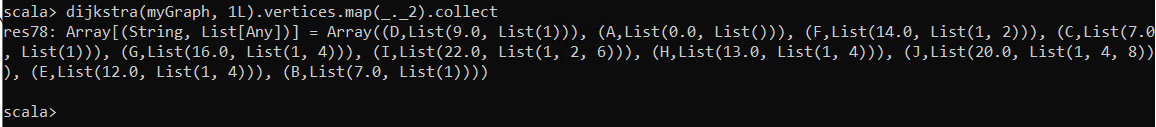
Автоматично згенерований опис





Зображення, що містить текст, знімок екрана, меню, Шрифт

Автоматично згенерований опис



1.7 Розв’язання задачі комівояжера за допомогою «жадібного» алгоритму

Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

Зображення, що містить знімок екрана, текст, Шрифт, ряд

Автоматично згенерований опис

1.8 Реалізація алгоритму Краскала

Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

**Висновки**

Було отримано навички роботи з графами з використанням Spark GraphX. Відпрацьовані завдання за прикладами.