Lab 02: MapReduce Programming

Instructor: Doan Dinh Toan
Department of Computer Science,
Faculty of Information Technology
University of Science - VNU-HCM
toandd.i81@gmail.com

Abstract

This lab assignment introduces students to the MapReduce programming model and provides practical experience with solving ten different problems using MapReduce. This assignment aims to help you develop your problem-solving skills and gain practical experience with the MapReduce programming model.

0 Preliminary

0.1 Reminder

The main objective of this course is to learn, and truly learn. You can discuss this with your classmate, but you need to take responsibility for your submission, which actually depends on your understanding of this course. For any kind of cheating and plagiarism, students will be graded 0 marks for the whole course.

0.2 Submission guideline

Each team submits its result to a folder named `teamABC`, with `ABC` being the team's name. The folder structure is as follows:

```
teamABC
├── src
│   ├── problem01
│   └── problem02...
├── docs
│   ├── report.md
│   ├── report.pdf
│   └── images
└── readme.md
```

- `src` is the folder for your source code. If the lab assignment is split into multiple problem, you have to save your script in a separate folder, corresponding to the given lab assignment.
- `docs` is the folder for your documents, including the work report and images associated with your report. If the lab assignment requires screenshots as proof, the images need to be stored in this folder if you inserted them in the report.
  - `report.md` is your report file in MarkDown format. The report must be written in English. This assignment will come with a template folder that already has a report template (you can use my OSCP template or create your own). If you are not familiar with MarkDown, see this cheat sheet. The report must include the following items:
    - ∗ Did you code by yourself or reference the solution?
    - ∗ Explain the code in detail.
    - ∗ Take screenshots of the running process and results.

    - * Self-reflection.
    - * References to your work.
  - report.pdf is the PDF file of your report, converted from the MarkDown file mentioned above.

- readme.md is the file that introduces your team and this lab assignment, this file should include the following basic information:

  1. Information about the course, the assignment, and notes to the instructors (if any).
  2. Information about your team (Student ID, full name of each member).

0.3 Rubrics

For each problem, students will get 1 point. If your do all the problem correctly without any references to the provided solution, you will get 1 point as bonus.

1 Instructions

Follow these instructions for each problem:

- Read the requirements of the problem and think about a solution.
- Try to solve the problem by yourself first.
- If you can solve the problem, write a MapReduce program to implement your solution. Include comments in your code to explain your thought process and any assumptions you made.
- Run your MapReduce program and capture the output. Include the output as part of your submission.
- If you are stuck or need help, reference external sources to get inspiration and guidance. However, you should always correctly cite your sources and not copy code or solutions without permission.
- Submit your solution code, instructions on how to run it, and the output.

Copying solutions without proper attribution is considered plagiarism and can result in serious consequences. Any copied solutions will receive a grade of 0 if there are no suitable references.

Some dataset in exercises could not be exist, so you can use other dataset to run your program. Some resources you can download dataset such as:

- PUMA Benchmarks
- EHPD
- Alteryx Designer Knowledge Base

You could search for other alternative datasets on the internet or generate a dataset to evaluate (please describe it in the report). Efficient algorithm design is crucial for processing large-scale graphs in the context of big data, you might found some interesting ideas in this work [2].

2 Problems

For the problem from 1 to 9, students checkout the detail requirements in the document [1], this document has beed uploaded to the drive folder of this lab.

2.1 Wordcount Program

2.2 WordSizeWordCount Program

2.3 WeatherData Program

2.4 Patent Program

2.5 MaxTemp Program

2.6 AverageSalary Program

2.7 De Identify HealthCare Program

2.8 Music Track Program

2.9 Telecom Call Data Record Program

2.10 Count Connected Component Program

2.10.1 Problem statement

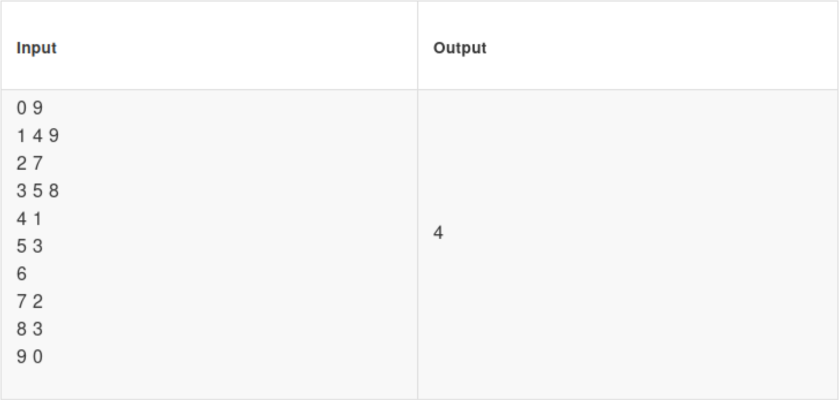
You are given an adjacency list representation of an undirected graph. Your task is to write a MapReduce program that counts the number of connected components in the graph.

2.10.2 Input

The input consists of a list of vertices and their adjacent vertices. Each vertex is represented by a unique number listed on the leftmost side of the input. The adjacent vertices of each vertex are listed after it, separated by a space.

2.10.3 Output

Number of conntected component in graph.



Input	Output

0 9
1 4 9
2 7
3 5 8
4 1
5 3
6
7 2
8 3
9 0
	4

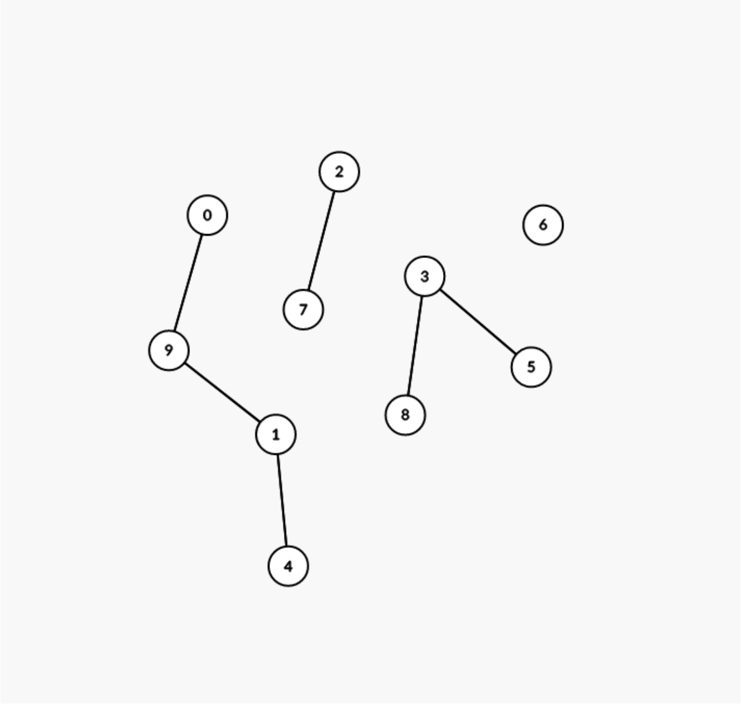


Figure 1: Visualization

References

[1] Sriram Balasubramanian. *Hadoop-MapReduce Lab*. University of California, Berkeley, 2016.

[2] Jimmy Lin and Michael Schatz. Design patterns for efficient graph algorithms in MapReduce. In *Proceedings of the Eighth Workshop on Mining and Learning with Graphs*, pages 78–85, Washington, D.C., July 2010. ACM.