Big programming assignment 1. Deadline 13.12.2023

Assume that the input is given as an *edge* relation where `Edge(a,b,x)` means that there is a directed edge in a graph between nodes `a` and `b` that has positive length `x`. Your task is to compute the *path* relation where `Path(a,b,x)` means that there is a non-empty sequence of edges starting from node `a` and ending in node `b` where the total length of edges in the sequence is `x` and there is no other path between those nodes that is shorter.

Examples:

1. Input:

`Edge(1,2,1), Edge(2,3,1), Edge(3,4,1)`

Expected output:

`Path(1,2,1), Path(2,3,1), Path(3,4,1), Path(1,3,2), Path(1,4,3), Path(2,4,2)`

2. Input:

`Edge(1,2,1), Edge(1,3,3), Edge(2,3,1)`

Expected output:

`Path(1,2,1), Path(2,3,1), Path(1,3,2).`

You need to implement and compare the following algorithms in Spark SQL:

- new paths are discovered by extending existing paths by one edge at a time,
- new paths are discovered by combining two existing paths.

Hints:

- Make sure you do not compute the same paths again and again.
- Caching, checkpointing and unpersisting data frames in appropriate moments, may be crucial for performance. In case of performance problems it is usually a good idea to inspect the lineage graphs.

You also need to test your solutions on the database of your choosing from https://snap.stanford.edu/data/ and see how large graphs can be processed on the cluster you set up in the GCP. You will be asked to present your solution and explain what are the advantages and disadvantages of particular variants of the algorithm.

For the sake of automatic testing your solution needs to follow the following steps:

- It should be a Python script that takes 3 command line arguments:
  1. Name of the algorithm: “linear” or “doubling”
  2. Path of the input file
  3. Path of the output file
- The input file will be a CSV (with the comma “,” as the separator and the header row) file containing 3 columns: edge_1, edge_2, length. Each row describes one element of the `Edge` relation.
- The program should write the computed `Path` relation to the SINGLE (not hdfs) output csv file, following the same format as the input file.
- The program should NOT read anything from the standard input. The standard output of the program will be discarded.