

## Algorithms

### Homework 4: due 19 June 2019

Subgraph matching (also known as subgraph isomorphism) is defined as follows: Given a data graph  $G$  and a query graph  $q$ , find all embeddings (i.e., matches) of  $q$  in  $G$ . See Figure 1. The state-of-the-art subgraph matching algorithm DAF finds matches by backtracking using a directed acyclic graph (DAG) of the query graph. Since many DAGs can be derived from the query graph, DAF chooses a rooted DAG among them and uses it. The performances of DAF may differ depending on which rooted DAG is chosen.

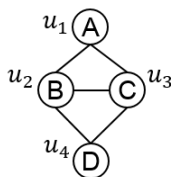
In this project, one team consists of 3 people, and you are to find a rooted DAG of the query graph which leads to good performances of DAF. You are to write a program that gets as input data graph  $G$  and query graph  $q$ , and outputs query DAG  $q_D$ . A query DAG can be represented by a linear order of the query vertices. For example, linear order  $(1, 3, 2, 4)$  represents a query DAG of the query graph in Figure 1a. After getting query DAG  $q_D$ , you can evaluate performances of DAF by giving  $G$ ,  $q$ , and  $q_D$  as input to DAF.

Since subgraph matching is an NP-hard problem, finding all matches may take too much time. In this project, therefore, DAF will find up to 100,000 matches of a query graph in a data graph. Also the time limit for a query graph is set to 1 minute. The following performance measures can be obtained for a query set that contains 100 query graphs.

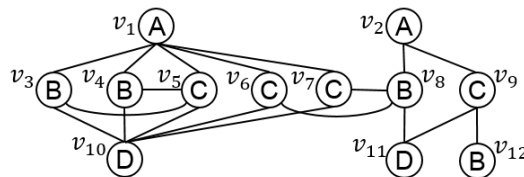
- average elapsed time: average running time for queries that are finished within time limit
- #recursive calls: average number of recursive calls for queries that are finished within time limit
- #solved queries: number of queries that are finished within time limit

A sample source code that makes a query DAG can be found in the following site (<https://github.com/SNUCSE-CTA/DAF>). You are to fork the repository in Github, and work by modifying the source code. If you have any questions about the program or homework, leave your questions on the issue board of the repository.

- Write in your report how you make a query DAG.
- Compare the performances of DAF with your query DAG and the original DAF in your report.
- Hand in your report and program by email to TA ([yychoi@theory.snu.ac.kr](mailto:yychoi@theory.snu.ac.kr)).
- Write down the environment you run your program and how to run your program in your report.
- Write comments appropriately in your program.



(a) Query graph  $q$



(b) Data graph  $G$

Figure 1: Query graph and data graph of subgraph matching.