Graph-Pattern-Matching-Challenge Report

Team : 2014-12940 김효건, 2018-16533 한승민
Date : 2021-06-08

1. Environment

C/C++: g++ (GCC) 6.3.0

2. How to run

```
mkdir build
cd build
cmake ..
make
./main/program <data graph file> <query graph file> <candidate set file>
```

3. Matching order and Backtracking.

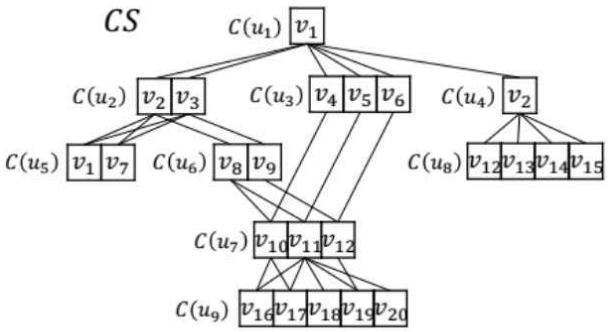
구현방식
1. Candidate Set에서 Candidate size가 작은 id 순으로 Matching order.
2. Matching order을 따라 backtracking을 이용한 search

Backtracking Algorithm

```
1. if (size of result_map == Query vertex size)
2.     then Print result_map
3. else
4.     for (next candidates)
5.         if (candidate is already used)
6.             then break, search other candidate
7.         if (candidate isn't connected)
8.             then break, search other candidate
9.         Add candidate to result_map, search next vertex
```

4. Example

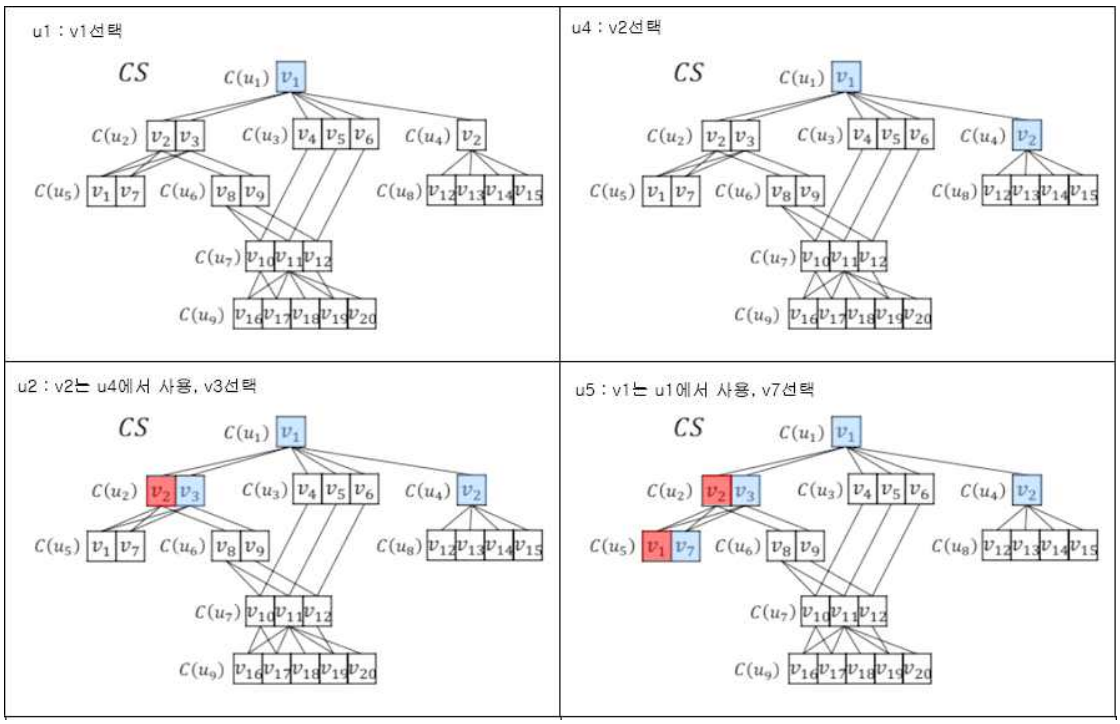
1. 다음과 같이 CS가 주어져있을 때,

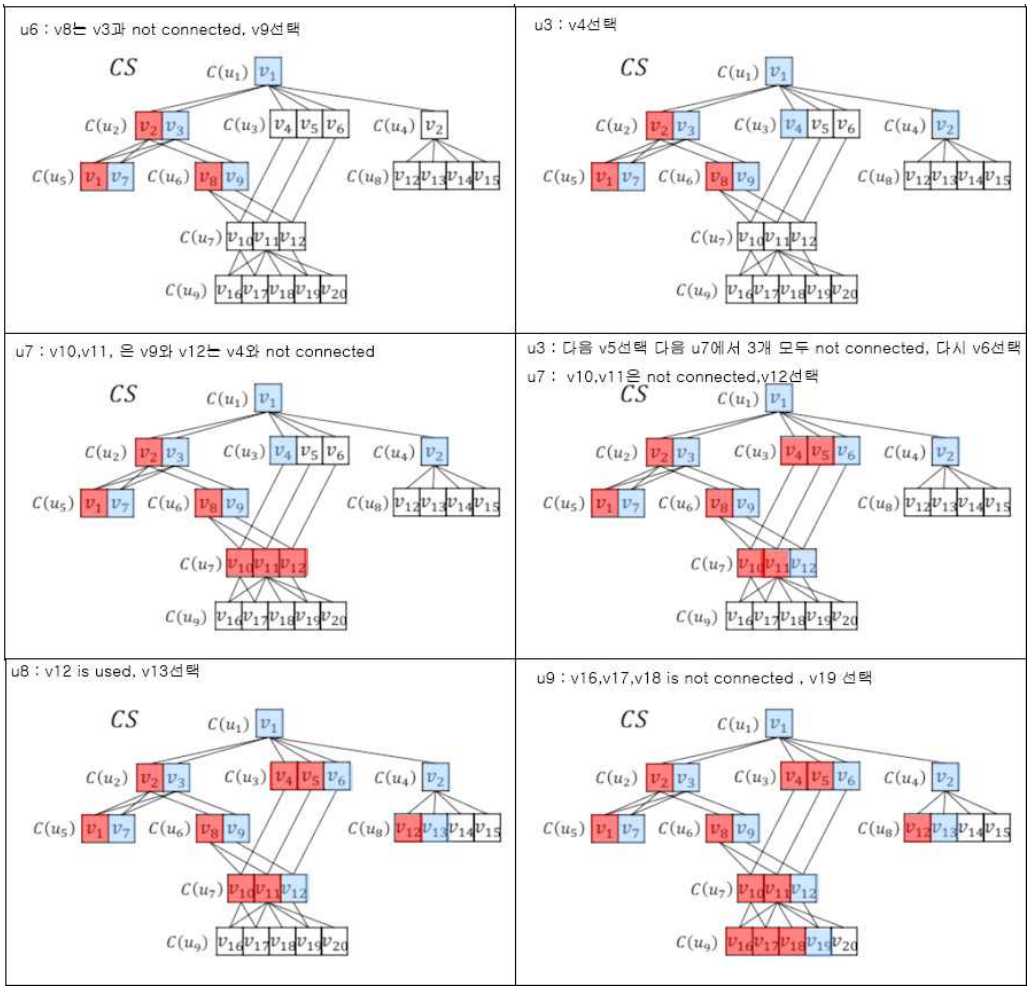


2. 개수가 작은 순서인 u1(1개) -> u4(1개) -> u2(2개) -> u5(2개) ->u6(2개) -> u3(3개) -> u7(3개) -> u8(4개)-> u9(5개) 순으로 검증

3. 각 검증 단계는
   1) 앞서서 사용된 vertex는 invaild
   2) Connected 되어져 있지 않으면 invaild

5. Example to get first result





6. Test Result

다음 환경에서 Terminal을 사용해 100,000개 출력으로 테스트 해본 결과

macOS Big Sur
Version 11.2.3

MacBook Pro (13-inch, 2019, Four Thunderbolt 3 ports)
Processor 2.4 GHz Quad-Core Intel Core i5
Memory 8 GB 2133 MHz LPDDR3
Startup Disk Macintosh HD
Graphics Intel Iris Plus Graphics 655 1536 MB

<Result>

Query	Time
lcc_hprd_n3.igraph	5.64 sec
lcc_hprd_n5.igraph	2.97 sec
lcc_hprd_n8.igraph	10.33 sec