

# Graph pattern matching challenge

2019-10011 이경훈

2019-16900 전종언

## I. 실행 환경

제공되어진 C++를 기반으로 <backtrack.cc>와 <common.h>를 수정하여 작성하였다.

실행 방법은 제공된 대로

```
[[
```

```
mkdir build
```

```
cd build
```

```
cmake ..
```

```
make
```

```
./main/program <data graph file> <query graph file> <candidate set file>
```

```
]]
```

를 이용하여 실행할 수 있다.

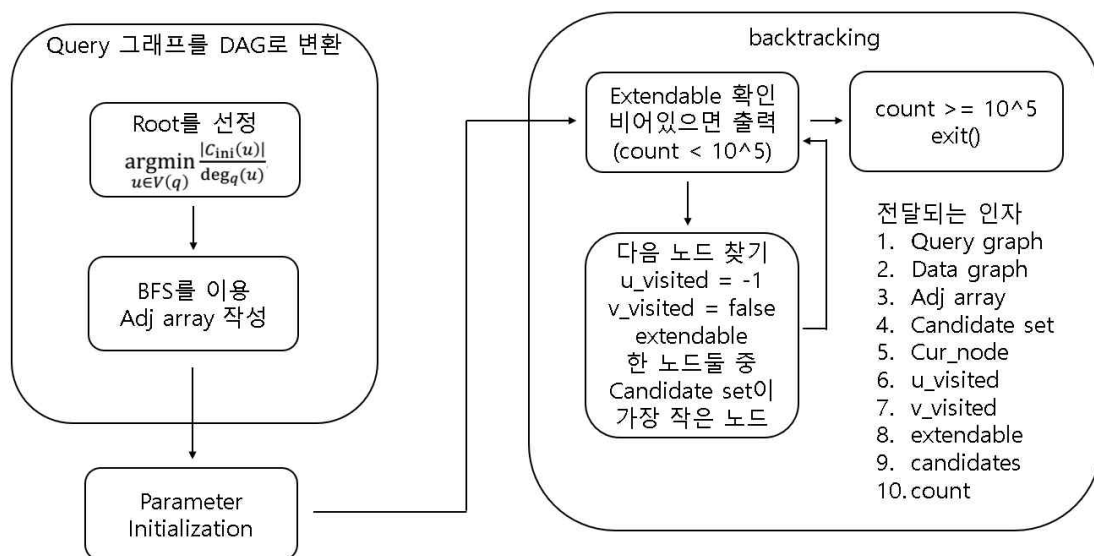
## 2. 작성 사항

2-1. <common.h>

#include <queue>를 추가하였다.

2-2. <backtrack.cc>

matching order의 경우 'Candidate-size order'를 이용했으며 backtrack과정을 포함한 전체 프로그램의 흐름은 아래 그림과 같다.



전달되는 인자 추가설명

u\_visited : Query graph에서 방문한 노드인지 저장, 방문하지 않았다면 -1, 방문했다면 v(u)

v\_visited : 사용된 candidate set의 v값인지 boolean형식으로 저장

extendable : 현재 노드(u)에서 갈 수 있는 다음 노드들의 후보군

candidates : 현재 노드에서 사용할 수 있는 v들의 후보군

count :  $10^5$ 개의 출력을 위해 카운팅하는 변수