

## **Algorithms**

# **Graph Pattern Matching Challenge**

2015-16227 김혜수

2017-17935 최형석

## 1. 실행 환경 및 실행 방법

사용 언어: C++

컴파일러: g++ (MinGW.org GCC-6.3.0-1) 6.3.0

컴파일 방법:

```
mkdir build
```

```
cd build
```

```
cmake ..
```

```
make
```

**실행 방법**

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

제공된 README.md 파일의 방법과 동일.

## 2. 실행 결과

제출된 코드로 모든 query에 대하여 프로그램을 실행한 결과를 나타내었으며, 이때의 채점 환경은 다음과 같다.

OS: windows

CPU: Intel(R) Core(TM) i7-10510U CPU @ 1.80GHz 2.30 GHz

Ram: 12GB

**실행 시간**

단위 second	n1	n3	n5	n8	s1	s3	s5	s8
hprd	0	3	1	5	0	2	4	4
human	0	18	2	60	0	1	0	1
yeast	1	9	3	7	4	60	60	60

**embedding 의 수**

단위 개	n1	n3	n5	n8	s1	s3	s5	s8
hprd	96	max	32,832	max	504	max	max	max
human	max	max	max	-	max	max	max	max
yeast	max	max	max	max	max	-	-	-

프로그램은 최대 100,000 개의 embeddings 를 출력하거나, 최대 60 초 동안 작동하도록 설정하였다. 총 24 가지의 query 중 60 초 미만의 실행 시간을 가지는 경우는 100,000 개 혹은 가지고 있는 embeddings 를 모두 출력한 것을 알 수 있으며, human\_n8, yeast\_s3, yeast\_s5, yeast\_s8 총 4 가지의 query graph 에 대한 embedding 은 찾지 못하였다.

### 3. Build DAG

Matching order 를 정하기 위하여, 주어진 query graph 를 바탕으로 DAG 를 만든다. 이때, query 의 root 는 query graph 의 vertices 중  $\frac{|C_{ini}(u)|}{deg_q(u)}$ 의 값이 최소화 되는 vertex 로 선정하였으며, 이는 Candidate 의 수가 가장 적으면서, degree 가 큰 vertex 를 root 로 선정하기 위함이다. 이후, 위 과정에서 선정된 root 를 이용하여 BFS 를 진행하며 source vertex 와 destination vertex 를 각각 src\_vv 와 dest\_vv 벡터에 저장한다.

### 4. Backtracking

Backtracking 은 앞선 과정에서 생성한 DAG 를 이용하여 진행하며, 다음 상황에 따라 재귀적으로 backtracking 을 한다.

**case 1.** src\_vector 의 크기가 query 의 vertex 수와 동일한경우  
query 의 모든 vertex 가 포함되었음을 의미하므로, 해당 embedding 을 출력한다.

**case 2.** srd\_vector 가 empty 인 경우  
root 에 대한 backtracking 시 실행되며, root 의 candidate 을 src\_vector 에 포함하며 재귀적으로 backtracking 을 호출한다.

**case 3.** 그 외.  
matching order 에 맞는 다음 vertex 를 선정하여 해당 vertex 의 candidate 을 srd\_vector 에 포함하며 재귀적으로 backtracking 을 호출한다.

## 5. Matching order

앞선 Backtracking 의 case 3 에서 다음 vertex 를 선정하는 로직에 해당한다. DAG 를 바탕으로 다음 candidate 가 가장 적은 vertex 를 선정한다. 이러한 matching order 를 바탕으로, backtracking 을 이어나가며 srd\_vector 의 크기가 query 의 vertex 와 동일해졌을 때, backtracking 을 종료하도록 한다.