

# 자료구조 프로젝트 1 보고서

2014170119  
신소재공학부 정준

- 선택한 자료구조
  - 리스트 (파이썬 기본)
  - 선택한 이유 : 구현이 간편했기 때문에. 일단 구현 후 다른 자료구조로 대체하려 하였다.
- 성능 예상
  - $u$  = 사용자,  $f$  = 친구 정보,  $t$  = 트윗

기능	성능 분석	
Read data files	$O(u) + O(f) + O(t)$	$O(u) + O(t)$
Display statistics	$O(u)(\text{for each users}) \times O(t)$ (TweetsByUser)	$O(u \times t)$
Top 5 most tweeted words	$O(t \log t)$ (python 기본 정렬)	$O(t \log t)$
Top 5 most tweeted users	$O(u) \times O(t)(\text{TweetsByUser}) \times O(u \log u)$ (python 기본 정렬)	$O(t \times u^2 \log u)$
Find users who tweeted a word	$O(t)$	$O(t)$
Find all people who are friends of the above users	$O(u')$ ( $u'$ =위 기능에서 선택된 사용자)	$O(u')$
Delete all mentions of a word	$O(t)$	$O(t)$
Delete all users who mentioned a word	$O(u)(\text{Transpose}) \times O(f) + O(u) \times O(f)$ $\times (O(u)(\text{UserByNum}) + O(t)$ (TweetsByUser))	$O(u^3) + O(u^2 \times t)$
Find strongly connected components	$2 \times (O(u)(\text{DFS}) + O(u \log u) (\text{pyton 기본 정렬})) + O(u) (\text{Transpose})$	$O(u \log u)$
Find shortest path from a given user	$O(u') + O(u') \times O(u' \log u')$ ( $u'$ =선택된 사용자와 연결된 모든 사용자)	$O(u'^2 \log u')$

- 성능을 개선할 수 있는 방법
  - 자료구조로 binary search tree를 사용하고 red-black tree를 적용한다.
  - 다익스트라 알고리즘에서 우선순위 큐를 사용한다.
  - 해시 테이블을 사용한다.