

문제해석

주어진 그래프의 부분 그래프 중 완전 그래프를 모두 찾는 문제

아이디어

부분 component가 k개의 node를 가진 완전 그래프이고,
component에 속하는 모든 node들이 하나의 node와 연결 돼 있을
때 k+1개의 node를 가진 완전 그래프가 생긴다

구현에 사용된 함수

- Find_common:** 주어진 부분 component의 모든 node들과
연결 돼 있는 node들을 모두 찾는 함수
- Generator:** 부분 component와 Find_common 함수로 찾은
모든 node 각 각을 조합하여 k+1개의 node를 가진 모든 완전
그래프를 만드는 재귀 함수

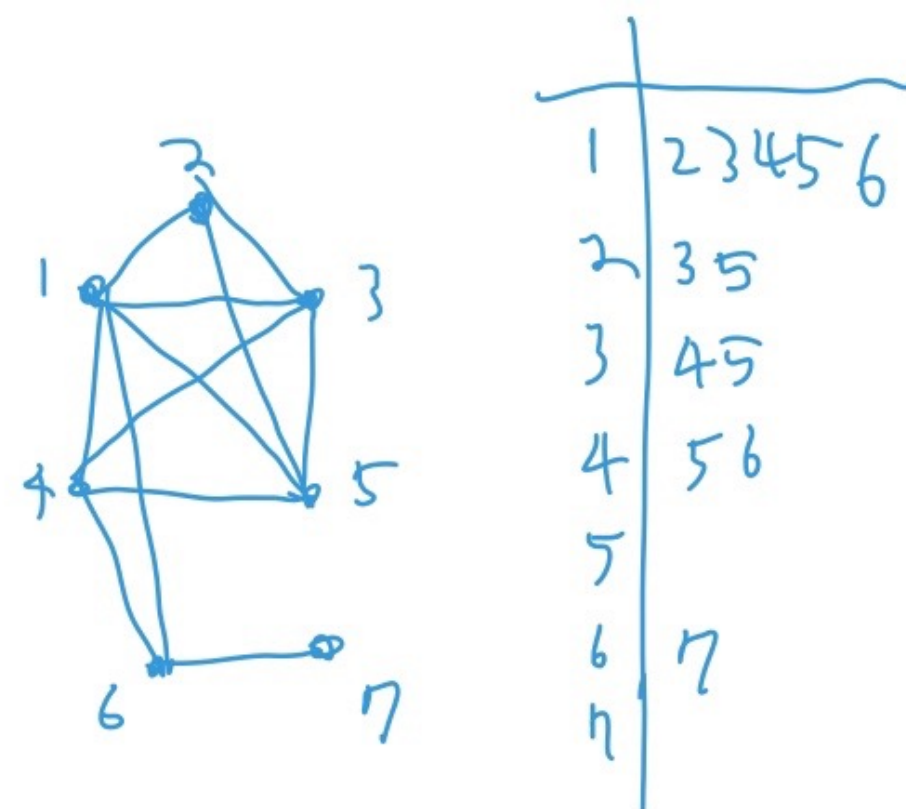
Find_common

부분 Component의 모든 node와 edge를 완전탐색하여 구현한다
Component의 모든 node보다 큰 id를 가진 node만 리턴한다

Generator

k개의 node를 가진 component를 입력받고, k+1개의 node를 가진
component들을 만들어 각 각 Generator를 호출한다.

k = 1인 경우로 시작하고, n개의 모든 node 각각에 대해 실행한다
Find_common으로 찾을 수 있는 node가 없을 때 종료한다.



예제의 모든 완전 부분 그래프

k = 2		k = 3	k = 4
(1,2)	→	(1,2,3) (1,2,5)	(1,2,3,5)
(1,3)	→	(1,3,4) (1,3,5)	(1,3,4,5)
(1,4)	→	(1,4,5) (1,4,6)	
(1,5)			
(1,6)			
(2,3)	→	(2,3,5)	
(2,5)			
(3,4)	→	(3,4,5)	
(3,5)			
(4,5)			
(4,6)			
(6,7)			

자료구조

그래프의 연결관계를 표현하는 인접리스트

각 node의 무게를 저장한 1차원 배열

부분 Component를 표현하는 1차원 배열

Find_common의 결과 set을 표현하는 1차원 배열

Find_common의 완전탐색에서 사용하는 1차원 배열

시간 복잡도

Find_common에서 componet의 전체 수 만큼
완전탐색을 반복하므로 $O(n * k)$ 의 비용이 든다.

Find_common은 Generator가 호출될 때마다
한 번씩 호출된다.

Generator는 완전 부분 그래프의 개수 만큼 호
출된다.

토의점

문제에서는 최대 450개의 노드와 900개의 edge가
주어진다. 42~43개의 노드가 900개의 모든 edge를
독점하는 경우 k가 42인 완전그래프가 그려진다.
이와 같이, 최대 k의 수가 큰 경우, 본 풀이로는 시간
내에 해결할 수 없을 것으로 예상된다