

C9. Tumor

박세찬, 이원빈

Problem Analysis

- n개의 노드와 b개의 간선, 각 노드 별로 weight가 주어졌다.
- 이 때, 가능한 fully connected Graph들의 weight 중 최대값을 구하여라.

Solution

- 1) 노드의 개수 n과 간선의 개수 b, 노드의 무게를 weight[n], 각 노드와 연결되어있는 노드들을 저장하는 벡터 `vector<int> link[n]`, 가능한 clique의 weight의 최대값을 maxWeight라고 하고, 0으로 초기화한다.
- 2) 한 노드를 포함하는 모든 클릭을 찾는 함수를 searchCliques라고 정의
 - a) 인자로 벡터 C, P, X를 받는다. C는 현재 찾고 있는 clique의 element, P는 현재 계산중인 노드와 연결되어 있는 노드들, X는 제외된 노드.
 - b) 처음 호출될 때는 C에 하나의 노드, P에는 그 노드와 연결되어 있는 노드들을 인자로 받는다.
 - c) 호출될 때 마다 C의 노드들의 weight를 더한 값을 maxWeight와 비교하여 maxWeight를 갱신
 - d) P의 노드들을 순회하며 X에 없으면서 C의 모든 노드와 연결되어 있는 노드 p가 있다면 searchCliques를 재귀호출
 - i) newC에 C의 노드와 p의 노드를 추가하여 인자로 전달, newP에는 p와 연결되어 있는 노드를 추가하여 인자로 전달.
 - e) P에 조건에 해당하는 노드들이 없다면 재귀호출을 종료.
- 3) 첫 번째 노드부터 n번째 노드까지 총 n번 searchCliques를 호출한다.
 - a) 이 때 그 노드의 함수 호출이 끝나면 그 노드의 중복계산을 피하기 위해 X에 추가한다.
- 4) 최종적으로 maxWeight를 출력한다.

Time Complexity

- 노드의 개수를 n, 간선의 개수를 b 라고 할 때, $O(n * b * 2^n)$ 이다.