P10

21700477 윤다은

문제정의

클러스터를 이루는 암세포중에서 가장 무거운 클러스터를 찾는 문제

아이디어

클러스터에 포함된 암세포의 개수를 k라고 할 때, 클러스터가 더 이상 발견되지 않을 때까지 k를 늘려가며 클러스터가 있는지 검사한다(재귀 활용).
만약 clique가 있다면 암세포의 무게를 합산하여 가장 큰 무게로 업데이트 한다.

세부구현

cell[n] : 현재까지 클러스터를 만족하는 암세포 배열
patri[n][n] : 암세포간의 결합유무를 저장하는 배열(adjacency matrix)
d[n] : 암세포마다 존재하는 혈관의 개수
weight[n] : 각 암세포의 무게를 저장하는 배열
keep : k 값을 늘릴지 말지 결정하는 변수(초기 keep 값 = true)

입력

암세포의 무게를 weight에 저장하고 암세포간의 연결을 1로 하여 patri에 기록한다. 이 때 암세포의 아이디에 따라 d의 값도 1씩 증가시킨다. 첫 재귀 호출 후 keep 값을 false로 한다.

재귀(경우의 수 탐색)

세 매개변수 : 시작 암세포의 인덱스(i), 현재 클러스터를 이룬 암세포의 개수(l), 원하는 클러스터의 크기(k)
cell에 들어가는 암세포의 아이디가 오름차순으로 하기 위해서 i를 두었다(중복방지).
$i+1 \le j \le n-(s-l)$인 j에 대해 다음을 반복한다.
cell[l]에 j를 대입한다.
만약 j가 클러스터를 형성할 수 있는 가능성이 있는 경우$(k-1 \le j)$
j를 포함한 cell이 클러스터를 만족하는지 확인한다.
만약 클러스터를 만족하고 $l < k$인 경우 새로운 클러스터를 탐색할 수 있으므로 j, l+1, k를 매개변수로 함수를 호출한다.
클러스터를 만족하고 $l = k$인 경우 cell에 저장되어 있는 암세포들의 무게를 합산하여 새로 계산한 무게가 더 무거울 경우 w를 갱신하고 keep을 true로 변경한다.
재귀를 마치고 돌아왔을 때 keep이 true라면 k를 1 증가하여, 재귀 과정을 다시 수행한다.
만약 false라면 w를 출력한다.

클러스터 검사

cell에 저장된 암세포간의 모든 쌍을 비교한다.만약 쌍중에 하나라도 연결되어 있지 않다면(patri == 0) false를 반환한다.

시간복잡도

입력 : O(n) - weight O(v) - patri입력
재귀 : k개에 대한 subgraph는 총 ${}_nC_k$이고, k에 대하여 클러스터 검사는 k^2의 시간이 걸린다
따라서 시간복잡도는 $O(n^k k^2)(1 \le k \le n)$이다

공간복잡도

모든 배열에 대해 n의 최대값인 450으로 선언했으므로 공간복잡도는 O(1)이다.

토의

k가 상수로 주어져있는 경우면 다항식 내에 문제가 해결될 수 있지만, 이 경우 k가 유동적으로 변하기 때문에 지수함수로까지 시간복잡도가 커지게 된다. 그럼에도 불구하고 통과한 경우는 k값이 테스트 케이스에서 아주 큰 값으로 가지 않았기 때문이 아닐까라는 생각을 하였다.