

# C1. Task Force

C1. Task Force문제는  $n$ 개의 marine와  $f$ 개의 relationships가 있는 team에서 모든 marine이  $k$ 개 이상의 relationship을 가지고 있는 sub-team의 최대 marine의 수를 찾는 문제이다.

해결책 방법은  $n$ 개의 1차원 배열을 만들어서,  $n$ 번째 marine이 가진 relationship의 개수를 저장. 2차원배열에 relationship의 유무를 1과 0으로 저장. 이후  $k$  미만 relationship의 개수를 가진 marine의 relation은 삭제 처리함. 즉, 해당 marine과 related된 marine의 relation을 1차원 배열에서 1씩 감소시킴. 2차원 배열에서도 서로 연결점을 1->0으로 수정. 위 과정을  $k$ 미만의 relationship이 1차원 배열에 없을 때 까지 반복한다.

코드의 내용은 2중 for문을 사용해 ID1부터  $n$ 까지 relationship을 저장한 2차원 배열을 확인 후  $k$ 보다 relationship이 적으면 ID번째 1차원 배열의 index value를 -1. 뒤 ID를 지을때 iterator가 이미 지난 ID의 relationship이  $k$ 보다 작아질 수 있기 때문에 그 부분은 재귀함수를 통해 해결. 시간적으로 처음 2차원 배열을 생성하는 부분에서 2f, 2중 for문 안에 연관되지 않은 변수로 1중 for문의 재귀 함수가 하나더 돌아가기 때문에,  $n^2 + a$ (재귀문이 돌아가는 횟수) $n + 2f$ 의 복잡도를 가진다.( $=O(n^2 + an + 2f) = O(n^2)$ )

공간적으로 relationship을 저장하는  $(n+1) \times (n+1)$ 의 int 2차배열 변수 및 전체 relationship의 개수를 저장하는 1차원 배열, input을 저장하는 배열, int 변수등을 사용했다. ( $=O(n^2)$ ) 하지만 굳이 relationship을 표현하는 extra 2차원 배열을 만들지 않고, input을 2차원 배열로 만들면, input을 저장하는 배열의 space를 아낄 수 있을 것이다.

위 문제에서 relationship을 1로 잡으면, biggest connected graph를 찾을 수 있다. 이를 응용한 문제는 biggest connected graph안의 도시중 임의의 A에서 B로 갈 수 있는 길의 존재여부를 파악하는 문제가 있다.