|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 3주차 - B 문제해결기법 보고서 | | | | | |
| 문제 | 조직개편 | | | | |
| 분반 | 002 | 학번 | 12141540 | 이름 | 박영창 |
| 알고리즘 개요 | * 사원의 수 n과 작업의 수 k를 입력 받는다. (1<=n<=100000) (1<=k<=300000) * Man 포인터 변수와 문서 정보를 가지는 int 변수를 멤버로 하는 Man 구조체 배열을 만든다. * 맨 처음 문서번호를 모두 1로, 1번 사원을 제외한 나머지 사원 구조체 내의 포인터 변수가 1번 사원의 주소를 가리키게 한다. * searchGoal 함수 : 두 개의 수를 입력 받고, 그 수를 인덱스로 사용하여 구조체 배열에 접근한다. 두 수에 해당하는 사원의 공통 조상과 깊이를 탐색한다. 두 사원의 깊이를 구하기 위해 각 사원의 위치에서 트리 구조로 봤을 때 1번 사원까지 위로 올라간다(포인터 변수를 타고 NULL까지 가고 그 횟수를 저장). 그 다음 깊이가 더 큰 사원의 깊이를 깊이가 더 작은 사원에게 맞춰 준다(트리를 올라간다). 두 사원의 깊이가 같아 졌을 때, 두 사원의 상사가 같은 지 확인하고 다르다면 두 사원 모두 트리를 올라가서 다시 확인한다. 같은 상사를 만날 때까지 반복한 뒤 미리 선언해둔 전역변수(두 깊이를 저장할 변수와 공통 조상의 포인터 변수)에 저장한다. O(logn) (최대:O(n)) * R=1의 경우   searchGoal 함수를 입력 받은 수를 통해 실행한다. 그 다음 두 수에 해당하는 사원의 깊이 중 더 깊이가 큰 사원의 문서 번호를 입력 받은 수로 바꿔주면서 깊이가 같아질 때까지 올라간다. 그 다음 두 사원의 공통 조상을 만날 때 까지 문서 번호를 바꿔주면서 올라간다. O(logn) (최대:O(n))   * R=2의 경우   두 개의 수를 입력 받고, 그 수를 인덱스로 사용하여 구조체 배열에 접근한다. 첫 번째 수에 해당하는 사원의 포인터 변수를 두 번째 수에 해당하는 사원의 주소 값으로 설정한다. O(1)   * R=3의 경우   searchGoal 함수를 입력 받은 수를 통해 실행한다. 문서 번호에 해당하는 배열 doc[101]을 선언한다. 그 다음 두 수에 해당하는 사원의 깊이 중 더 깊이가 큰 사원의 문서 번호를 확인해 문서 번호가 i라면 doc[i]를 증가시킨다. 깊이가 같아졌다면 두 사원의 공통 조상을 만날 때 까지 같은 작업을 반복한다. 그 다음 반복문을 통해 doc[i] (1<=i<100)를 모두 확인해서 0보다 크다면 kinds라는 변수를 증가시키고 반복문이 끝난 뒤 kinds를 출력한다. O(logn) (최대:O(n)) | | | | |
| 성능분석 | 입력 받은 사원 수 : n  searchGoal, document, kinds : O(logn), 최악의 경우 O(n)  under : O(1) | | | | |
| 실험결과 | 소요시간 : 0.21s  사용된 메모리 공간 : 1.7MB | | | | |
| 비고 |  | | | | |