|  |  |  |  |
| --- | --- | --- | --- |
| 서약서: 나는 본 과제를 수행함에 있어서 허가되지 않은 도움을 주거나 받지 않았음을 서약합니다. | | | |
| 분반:42 | 학번:2016312761 | 이름:여혁수 | (서명)여혁수 |
| 강의 시간 첨삭 자료로 활용함에 동의합니다. | | 동의함  ㅇ | 동의하지 않음 |

**분석 리포트 [과제명: 두더지 잡기! ]**

1. 분석(문제 분석, 자료 구조 및 알고리즘 분석)

문제 분석) 가장 먼저 사라지는 두더지의 종류가 무엇인지 구해야 하므로 각 종류별 두더지 개체 수를 저장해서 두더지를 잡을 때마다 1씩 감소시키거나, 두더지를 잡을 때마다 남은 두더지들의 종류를 탐색할 수도 있다. 최대 200만 마리의 두더지가 있기 때문에 두더지를 잡을 때마다 남은 두더지를 탐색하는 것은 시간 면에서 손해가 클 것 같다. 1부터 N까지 종류가 있다고 했으므로 N의 값을 알아내고, 1~N까지의 종류별 두더지의 개수가 파악이 되어야 먼저 사라지는 두더지의 종류를 상대적으로 빠르게 구할 수 있을 것 같다.

자료 구조) 1~N까지 종류별 두더지의 개수를 저장할 배열이 필요한데, N을 우선 알아내야 필요한 배열의 크기를 알 수 있으므로 동적으로 할당한다. 공간복잡도는 O(N)이 될 것이다.

알고리즘 분석) 일단 문제 조건에 따라 처음으로 잡을 두더지는 두더지 배열에서 맨 앞 칸 두더지이고, 그 다음부터 잡을 두더지는 직전에 잡았던 두더지에 쓰여진 숫자만큼 건너 뛰었을 때 보이는 두더지이다.

잡은 두더지의 해당 배열 칸을 0으로 만들고 다음으로 잡을 두더지를 찾는다. 일반적으로 다음 두더지의 위치는 직전에 잡은 두더지의 인덱스에 그 두더지한테 쓰여진 수를 더한 값이 가리키는 인덱스가 된다. 잡고 찾고를 반복하다가 배열 끝까지 탐색을 했다면 다시 첫 칸으로 돌아가야하는데, 이 때는 탐색하는 두더지들의 범위를 줄이기 위해 값을 0으로 셋팅했던 칸들을 제거한다. 제거하는 방법은 배열을 처음부터 끝까지 보면서 0인 값은 skip하고 아닌 것만 다시 배열의 첫 칸부터 담는 것이다. 그리고 다음 탐색에서도 배열의 끝을 알아야하므로 맨 끝 두더지 인덱스를 배열의 사이즈로 재설정한다. 그리고 나서 다음으로 잡을 두더지를 찾을 때, 배열이 재배열되었으므로 일반적인 상황이랑은 다른데, 그 이유는 직전에 잡은 두더지의 인덱스가 변했을 것이기 때문이다. 재배열 전의 배열을 생각해보면 직전에 잡은 두더지 이후로는 두더지를 잡지 않았기 때문에 값이 0인 두더지가 없을 것이고 그 이전에만 값이 0인 두더지가 있을 것이다. 따라서 직전에 잡은 두더지 인덱스에서 0이었던 두더지의 수를 빼주어야 재배열 시 알맞은 인덱스가 된다. 이렇게 인덱스를 변환하고 나서 다음으로 잡을 두더지의 인덱스를 구하는 수식은 변환한 인덱스와 건너뛰어야 하는 두더지 수를 더하고 재설정된 배열의 사이즈와 나머지 연산을 한 것이 된다. 사이즈를 초과하면 그 때부터 인덱스 0부터 시작하기 때문이다.

그리고 두더지를 잡을 때마다 종류별 두더지의 개체수를 저장하는 배열에서 해당 종류의 값을 1 감소시킨다. 감소시키기 전에 저장되어있는 개체수 값이 1이라면 감소시키면 0이 될 것이므로 그 종류를 리턴하면서 함수가 종료된다. 시간복잡도에서 최악의 경우는 정확한 예시를 상상하기는 어렵지만 배열을 재배열하고 두더지를 제거해야 할 때 두더지 수 만큼 탐색을 해야하고, 그 과정을 최대 (두더지 수 - 1) 만큼 반복해야 하는 경우일 것이다. 이 때 시간 복잡도는 O(m^2) {m: 두더지의 수} 이 된다.

2. 비고 (교수나 조교에 하고 싶은 말)

처음에는 잡은 두더지가 해당하는 배열 칸의 값을 0으로 바꿔서 잡은 두더지라는 것을 표시했는데, Timeout이 조금 떠서 시간복잡도를 줄이려고 잡은 두더지가 해당하는 칸의 앞부분이나 뒷부분을 당겨서 배열 범위를 1만큼 좁힘으로서 남은 두더지의 배열 범위 안에는 0인 값이 없게끔 만들었습니다. 이 과정이 좀 어려웠습니다. 하지만 0인 값을 없애니까 다음으로 잡을 두더지를 한 칸 씩 옮겨가며 찾을 필요 없이 한 번에 찾을 수 있어서 인풋 길이가 큰 경우에(10000이상) 꽤나 빨라지는 효과를 얻을 수 있었습니다. 그래도 Timeout은 피하지 못했습니다. 많이 헤맸던 문제였습니다.