초등부 4. 사탕 돌리기

풀이 작성자: 구재현

부분문제 1 ($Q = 1$)

다음과 같은 그리디 알고리즘을 사용할 수 있다.

- 현재 깡통에 있는 사탕 중 다른 위치로 옮겨져야 하는 사탕을 아무거나 고른다. 그런 것이 없다면 아무거나 고른다.
- 해당 사탕이 목적지에 도달할 때까지 계속 가지고 다닌다.

이러한 그리디 알고리즘을 구현했을 때 원하는 최종 상태가 나왔다면 문제를 해결할 수 있다. 알고리즘의 정당성은 따로 증명하지 않으나, 전체 문제 풀이에서 자연스럽게 따라온다.

부분문제 2 ($N = 2$)

1번 깡통에 있는 2번 사탕의 개수와, 2번 사탕에 있는 1번 사탕의 개수는 동일하다. 이 숫자를 C 라고 하자. 한 번의 사탕 돌리기로 두 사탕을 바꿔주면, C 를 하나씩 줄여줄 수 있다. 고로 $C \leq Q$ 면 답은 1, 아니면 0이다.

부분문제 5

먼저, 최종 상태를 만들 수 있는 최소 작업 수를 Q_{min} 이라고 하면, $Q_{min} + 1, Q_{min} + 2 \ldots$ 번의 작업으로도 원하는 상태를 얻을 수 있다. Q_{min} 번의 작업 이후에, 1번 깡통에서 아무 사탕이나 집은 후 한 바퀴 도는 식으로 남은 작업을 처리하면 되기 때문이다. 고로 최소 작업 수인 Q_{min} 을 구한 후 이것이 Q 이하인지를 판별하는 식으로 문제를 해결한다.

어떠한 사탕이 목표로 하는 깡통에 들어가 있지 않을 경우 이 사탕을 *어긋난 사탕* 이라고 정의하자. 큰 틀에서, 대략 다음과 같은 그리디 알고리즘을 사용할 수 있다.

- 현재 깡통에 있는 사탕 중 다른 위치로 옮겨져야 하는 사탕을 아무거나 고른다.
- 해당 사탕이 목적지에 도달할 때까지 계속 가지고 다닌다.
- 모든 사탕이 제 자리에 갈 때까지 반복한다.

이 알고리즘의 가장 큰 문제점은 현재 깡통에 있는 사탕 중 다른 위치로 옮겨져야 하는 사탕이 없을 수 있다는 것이다. 일단은 운이 좋게 그러한 일이 절대 일어나지 않는다고 가정한 후 논리를 전개하자.

첫 번째 관찰은, 어긋난 사탕을 제자리에 놓았을 때, 해당 깡통에 어긋난 사탕이 하나 이상 존재하거나, 현재 위치가 1이다. 만약에 현재 위치가 1이 아니면서 어긋난 사탕이 존재하지 않는다고 한다면, 제자리에 놓게 될 사탕을 포함하여 총 $K + 1$ 개의 사탕이 해당 위치에서 어긋나지 않게 되는데, 같은 색을 가진 사탕이 K 개이기 때문에 이는 모순이다. 고로, 사탕을 제 자리로 돌리는 과정은 하나의 사이클을 이룬다. 이를 *정리 사이클* 이라고 부르자.

두 번째 관찰은, 앞서 한 가정을 유지했을 때 이 알고리즘이 최적이라는 것이다. 어떠한 사탕 x 가 다른 위치 y 로 옮겨지기 위해서는, $x \leq y$ 일 때 최소 $y - x$ 번 꺼내져야 하고, 그렇지 않을 경우 $N - x + y$ 번 꺼내져야 한다. 이 사실은 어떠한 전략을 사용하더라도 피할 수 없으니, 우리가 사탕을 꺼내야 하는 횟수의 최소 한도는 정해져 있다고 할 수 있다. 한편 위 알고리즘은 항상 이 꺼내야 하는 횟수의 최소 한도만큼의 연산을 사용한다. 고로 위

알고리즘은, 나쁜 경우가 없다는 가정 하에, 최적이다.

이제 전체 알고리즘으로 넘어가자. 현재 내가 있는 위치에서 어긋난 사탕이 존재한다면, 우리는 *정리 사이클* 을 한 바퀴 돌면서 남은 사탕들을 최적의 연산으로 정리해 줄 수 있다. 어긋난 사탕이 없지만 다른 위치에 있다면, 해당 위치에서도 정리 사이클을 구해주자. 우리는 고로 사탕들을 여러 개의 최적인 정리 사이클로 분해할 수 있다. 여기서 세 번째 관찰은, 두 개의 정리 사이클을 하나의 정리 사이클로 합쳐줄 수 있다는 것이다. 정리 사이클 A, B 가 있을 때, B가 사탕을 교환하는 임의의 깡통에서 멈춰서, (A의 사탕 내려놓기) $\rightarrow$ (B의 사탕 집고 B의 정리 사이클 완료) $\rightarrow$ (다시 A에서 집으려 했던 사탕, 혹은 내려놓은 사탕을 집은 후 A의 정리 사이클 완료) 의 요령으로 두 사이클을 한번에 처리할 수 있다.

이렇게 되면 우리는 정리 사이클이 1번 깡통에서 사탕을 교환한다는 가정 하에 최적의 알고리즘을 찾았다. 모든 사탕들의 원호상 거리를 모두 더한 후, 이를 N 으로 나누면 된다. 1번 깡통에서 정리 사이클이 사탕을 교환하지 않는다는 것은, 1번 깡통에 1번 사탕만이 존재한다는 것이다. 이 사탕들에 대해서는 낭비를 피할 수가 없으니 답을 1 늘리는 것은 불가피하며, 아무 사탕이나 집고 정리 사이클을 만난 후, 정리 사이클을 처리하고 다시 그 사탕을 1번 깡통에 넣는 식으로 $\frac{\text{거리합}}{N} + 1$ 번 안에 정리가 가능하다. 이제 전체 문제는 입력으로 주어진 사탕에 대해서 거리 합을 구한 후, 단순 수식으로 쉽게 구현할 수 있다. 시간 복잡도 $O(NK)$, 공간 복잡도 $O(1)$ 에 문제가 해결된다.

부분 문제 3은 정리 사이클로 사탕을 분해하는 방법이 유일하여, 임의의 방법으로 분해할 생각을 하지 않아도 괜찮다. 부분 문제 4는 이 풀이와 다른 접근을 사용하는 $O(N^2K)$ 알고리즘을 위해서 주어졌다. 해당 알고리즘에 대한 설명은 생략한다.