

커밋 SHA: [cbaea4b3eb90b6d021616f8276657ebcc749353c](#)

커밋 메시지: [level 1] Title: 햄버거 만들기, Time: 1012.09 ms, Memory: 93.8 MB - BaekjoonHub

커밋 분석 결과:

공부했던 내용 : 프로그래머스의 133502번 문제인 "햄버거 만들기" 문제를 풀기 위해 JavaScript의 배열 조작 및 reduce 함수 사용법을 공부했다. 문제의 조건을 충족하는 효율적인 알고리즘을 설계하는 방법과, 배열에서 특정 패턴을 찾고 처리하는 방법을 익혔다. 특히, reduce 함수를 이용하여 배열을 순회하면서 동시에 누적 결과를 생성하고, 필요에 따라 배열의 요소를 추가하거나 제거하는 기법을 중점적으로 학습했다. 문제에서 제시된 제한사항(배열 길이 최대 1,000,000)을 고려하여 시간 복잡도를 최소화하는 알고리즘을 설계하는 데 집중했다.

작성된 기능 : 주어진 정수 배열 `ingredient`를 입력받아, "빵-야채-고기-빵" 순서로 햄버거가 완성되는 횟수를 계산하는 함수 `solution`을 작성했다. `reduce` 함수를 이용하여 배열을 순회하면서, 현재까지 누적된 재료 목록의 마지막 네 개의 요소가 "1231" (빵-야채-고기-빵)인지 확인한다. 만약 일치한다면, 해당 네 개의 요소를 목록에서 제거하고 햄버거 개수를 증가시킨다. 최종적으로 햄버거의 개수를 반환한다. 함수는 배열 조작을 위해 `push`, `slice`, `splice` 메서드를 사용하며, 함수형 프로그래밍 스타일로 작성되어 가독성을 높였다. `reduce`의 callback 함수 내부에서 객체 리터럴을 사용하여 추가적인 변수 선언 없이 햄버거 개수를 카운트한다.

아쉬웠던 부분 : 코드의 가독성을 더욱 높일 수 있었을 것이다. 특히, `reduce` 함수 내부의 callback 함수는 다소 복잡하고 이해하기 어려울 수 있다. `reduce`를 사용하지 않고, 더 명확한 방식(예: `for` loop)으로 작성하면, 코드의 가독성을 높이고 디버깅을 용이하게 할 수 있었을 것이다. 또한, 함수의 이름(`solution`)은 너무 일반적이며, 함수가 수행하는 기능을 명확하게 나타내는 이름(예: `countHamburgers`)으로 변경하는 것이 좋았을 것이다. 코드의 성능 측면에서는 `reduce` 함수를 사용하여 효율적인 코드를 작성했지만, 입력 배열의 크기가 매우 클 경우, 더욱 최적화된 알고리즘을 고려해 볼 수 있다. 예를 들어, 문자열 비교 대신 정수 비교를 통해 연산 속도를 향상시킬 수 있을 것이다. 더 나아가 코드에 대한 주석을 추가하여 각 부분의 동작을 명확하게 설명하는 것이 좋다.