

유형	Dynamic Programming	소요 시간	30분						
문제	<p>[문제 설명]</p> <p>길 위에 사과나무와 오렌지 나무가 있습니다. 사과 열매 하나당 1점, 오렌지 열매 하나당 -1점이라고 할 때, 길 위의 연속으로 이어진 나무에서 얻을 수 있는 최고 점수를 구하는 함수, solution을 완성해주세요.</p> <p>예를 들어, 길 위에 있는 나무의 열매 중 사과 열매의 수는 양수로, 오렌지 열매 수는 음수로 표현된 fruits [-2, 5, -3, 6, 8, -1, -5, 3]가 있을 때 얻을 수 있는 최고 점수는 2번째부터 5번째까지 수확했을 때 점수인 16점(5 - 3 + 6 + 8) 입니다.</p> <p>[제한 사항]</p> <ul style="list-style-type: none">- 열매 수확은 띄엄띄엄할 수 없으며, 연속된 나무의 모든 열매를 수확해야 합니다.- 사과 열매 하나는 1점, 오렌지 열매 하나는 -1점입니다.- 최소한 하나의 나무의 열매는 수확해야 합니다. <p>[입력 형식]</p> <ul style="list-style-type: none">- 길 위에 과일의 점수를 나타낸 fruits가 주어집니다.- fruits는 -1000 이상 1000 이하의 정수로 이루어진 배열입니다.- 길 위의 나무의 수는 1개 이상 10,000개 이하입니다. <p>[출력 형식]</p> <p>길 위의 연속으로 이어진 나무에서 얻을 수 있는 최고 점수를 출력합니다.</p>								
	<table><tr><th colspan="2">매개변수</th><th>Return</th></tr><tr><td>number[]</td><td>fruits</td><td>number</td></tr></table>			매개변수		Return	number[]	fruits	number
	매개변수		Return						
	number[]	fruits	number						
	<pre>/** * @param fruits {number[]} * @return {number} */ function solution(fruits) { var answer = 0; return answer; }</pre>								
<pre>function solution(fruits) { var n = fruits.length; // n번째 까지의 나무에서 얻을 수 있는 최고 점수 var dp = []; dp[0] = fruits[0]; // 현재까지 최고 점수 var curMax = dp[0];</pre>									

	<pre>for (var i = 1; i < n; i++) { dp[i] = fruits[i] + (dp[i - 1] > 0 ? dp[i - 1] : 0); curMax = Math.max(curMax, dp[i]); } return curMax; }</pre>																																				
테스트 케이스	<div>예시케이스(1개)</div> <table><tr><th>입력</th><th>출력</th></tr><tr><td>fruits</td><td></td></tr><tr><td>[-2, 5, -3, 6, 8, -1, -5, 3]</td><td>16</td></tr></table> <div>정확성 테스트케이스(5개)</div> <table><tr><th>입력</th><th>출력</th></tr><tr><td>fruits</td><td></td></tr><tr><td>[-1, -9, -9, -1, 3, 26, -9, 0, -3, 8, -9, 8, -6, 5]</td><td>29</td></tr></table> <table><tr><th>입력</th><th>출력</th></tr><tr><td>fruits</td><td></td></tr><tr><td>[-10, -9, -8, -7, -6, -5, -4, -3, -2, -1]</td><td>-1</td></tr></table> <table><tr><th>입력</th><th>출력</th></tr><tr><td>fruits</td><td></td></tr><tr><td>[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]</td><td>55</td></tr></table> <table><tr><th>입력</th><th>출력</th></tr><tr><td>fruits</td><td></td></tr><tr><td>[-1000, 1000, -1000, 1000, -1000, 1000, -1000]</td><td>1000</td></tr></table> <table><tr><th>입력</th><th>출력</th></tr><tr><td>fruits</td><td></td></tr><tr><td>[-1000, 2000, -3000, 4000, -5000, 4000, -3000, 2000, -1000]</td><td>4000</td></tr></table>	입력	출력	fruits		[-2, 5, -3, 6, 8, -1, -5, 3]	16	입력	출력	fruits		[-1, -9, -9, -1, 3, 26, -9, 0, -3, 8, -9, 8, -6, 5]	29	입력	출력	fruits		[-10, -9, -8, -7, -6, -5, -4, -3, -2, -1]	-1	입력	출력	fruits		[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]	55	입력	출력	fruits		[-1000, 1000, -1000, 1000, -1000, 1000, -1000]	1000	입력	출력	fruits		[-1000, 2000, -3000, 4000, -5000, 4000, -3000, 2000, -1000]	4000
입력	출력																																				
fruits																																					
[-2, 5, -3, 6, 8, -1, -5, 3]	16																																				
입력	출력																																				
fruits																																					
[-1, -9, -9, -1, 3, 26, -9, 0, -3, 8, -9, 8, -6, 5]	29																																				
입력	출력																																				
fruits																																					
[-10, -9, -8, -7, -6, -5, -4, -3, -2, -1]	-1																																				
입력	출력																																				
fruits																																					
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]	55																																				
입력	출력																																				
fruits																																					
[-1000, 1000, -1000, 1000, -1000, 1000, -1000]	1000																																				
입력	출력																																				
fruits																																					
[-1000, 2000, -3000, 4000, -5000, 4000, -3000, 2000, -1000]	4000																																				