

JavaScript 투 포인터 투 포인터 알고리즘 이해하기

투 포인터 알고리즘 이해하기 | 두 개의 점을 활용하는 알고리즘
강사 나동빈

JavaScript

투 포인터

투 포인터 알고리즘 이해하기

JavaScript 투 포인터 투 포인터 알고리즘

투 포인터 이해하기

JavaScript 투 포인터

투 포인터 이해하기

- 투 포인터 알고리즘은 다음과 같다.
- 리스트에 순차적으로 접근해야 할 때 두 개의 점의 위치를 기록하면서 처리하는 알고리즘을 의미한다.
- 흔히 2, 3, 4, 5, 6, 7번 학생을 지목해야 할 때 간단히 '2번부터 7번까지의 학생'이라고 부르곤 한다.
- 리스트에 담긴 데이터에 순차적으로 접근해야 할 때는 시작점과 끝점 2개의 점으로 접근할 데이터의 범위를 표현할 수 있다.

특정한 합을 가지는 부분 연속 수열 찾기

- $N = 8$ 개의 자연수로 구성된 수열이 있다.
- 합이 $M = 5$ 인 부분 연속 수열의 개수를 구하여라.
- 수행 시간 제한은 $O(N)$ 이다.

Element	1	2	3	4	5	6	7	8
Value	3	2	4	1	2	2	1	5

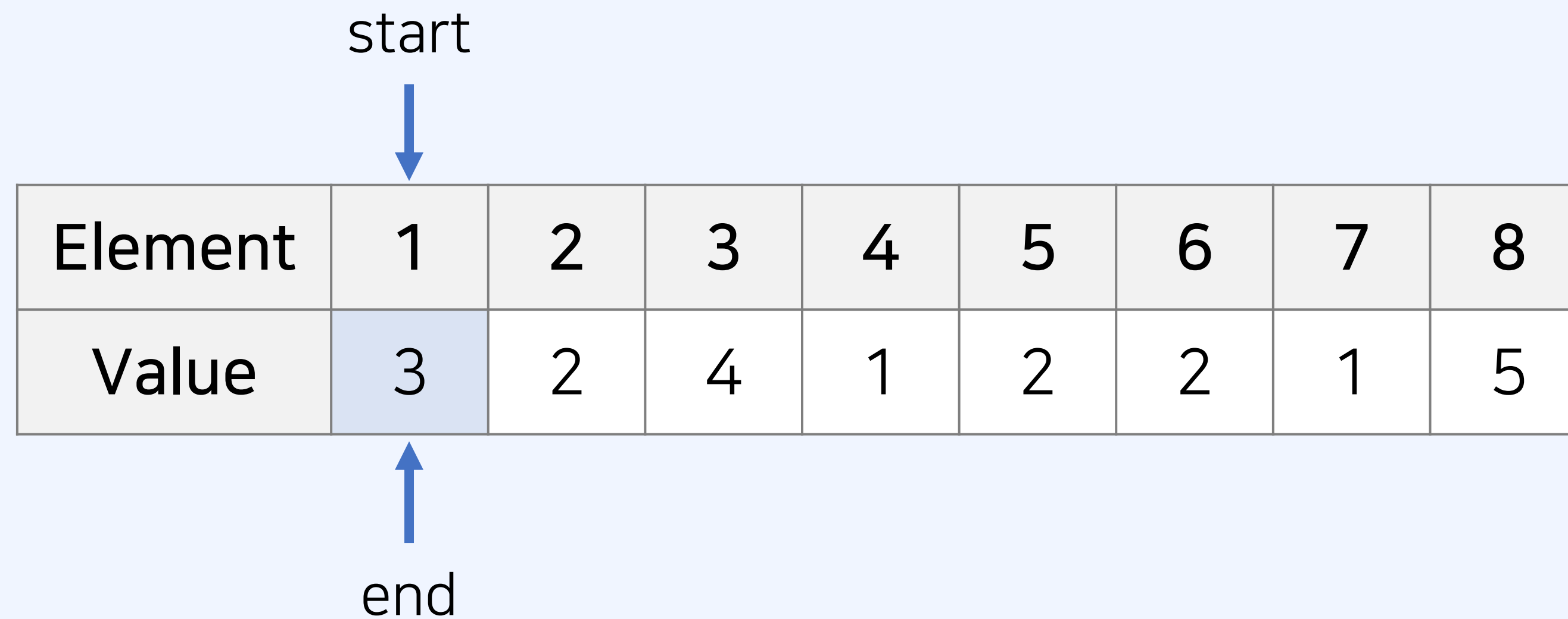
특정한 합을 가지는 부분 연속 수열 찾기

- 투 포인터를 활용하여 다음과 같은 알고리즘으로 문제를 해결할 수 있다.
 1. 시작점(*start*)과 끝점(*end*)이 첫 번째 원소의 인덱스(0)를 가리키도록 한다.
 2. 현재 부분 합이 *M*과 같다면, 카운트한다.
 3. 현재 부분 합이 *M*보다 작거나 같다면, *end*를 1 증가시킨다.
 4. 현재 부분 합이 *M*보다 크다면, *start*를 1 증가시킨다.
 5. 모든 경우를 확인할 때까지 2번부터 4번까지의 과정을 반복한다.

Element	1	2	3	4	5	6	7	8
Value	3	2	4	1	2	2	1	5

- Current sum = 3
- Count = 0

start



Element	1	2	3	4	5	6	7	8
Value	3	2	4	1	2	2	1	5

end

- Current sum = 5
- Count = 1

start

↓

Element	1	2	3	4	5	6	7	8
Value	3	2	4	1	2	2	1	5

↑

end


JavaScript 투 포인터 투 포인터 이해하기

특정한 합을 가지는 부분 연속 수열 찾기

JavaScript
투 포인터
투 포인터
이해하기


- Current sum = 9
- Count = 1

start



Element	1	2	3	4	5	6	7	8
Value	3	2	4	1	2	2	1	5

end



- Current sum = 6
- Count = 1

start
↓

Element	1	2	3	4	5	6	7	8
Value	3	2	4	1	2	2	1	5

↑
end

- Current sum = 4
- Count = 1

			start					
			↓					
Element	1	2	3	4	5	6	7	8
Value	3	2	4	1	2	2	1	5
			↑					
			end					

- Current sum = 5
- Count = 2

			start					
			↓					
Element	1	2	3	4	5	6	7	8
Value	3	2	4	1	2	2	1	5
				↑				
				end				

특정한 합을 가지는 부분 연속 수열 찾기

- Current sum = 7
- Count = 2

			start					
			↓					
Element	1	2	3	4	5	6	7	8
Value	3	2	4	1	2	2	1	5
					↑			
					end			

특정한 합을 가지는 부분 연속 수열 찾기

- Current sum = 3
- Count = 2

				start				
				↓				
Element	1	2	3	4	5	6	7	8
Value	3	2	4	1	2	2	1	5
					↑			
					end			

- Current sum = 5
- Count = 3

				start				
				↓				
Element	1	2	3	4	5	6	7	8
Value	3	2	4	1	2	2	1	5
						↑		
						end		

특정한 합을 가지는 부분 연속 수열 찾기

- Current sum = 6
- Count = 3

				start				
				↓				
Element	1	2	3	4	5	6	7	8
Value	3	2	4	1	2	2	1	5
							↑	
							end	

- Current sum = 5
- Count = 4

					start			
					↓			
Element	1	2	3	4	5	6	7	8
Value	3	2	4	1	2	2	1	5
							↑	
							end	

- Current sum = 10
- Count = 4

start
↓

Element	1	2	3	4	5	6	7	8
Value	3	2	4	1	2	2	1	5

↑
end

- Current sum = 8
- Count = 4

						start		
						↓		
Element	1	2	3	4	5	6	7	8
Value	3	2	4	1	2	2	1	5
								↑
								end

- Current sum = 6
- Count = 4

							start	
							↓	
Element	1	2	3	4	5	6	7	8
Value	3	2	4	1	2	2	1	5
								↑
								end

JavaScript 투 포인터 투 포인터 이해하기

특정한 합을 가지는 부분 연속 수열 찾기

JavaScript 투 포인터 투 포인터 이해하기

- Current sum = 5
- Count = 5

Element	1	2	3	4	5	6	7	8
Value	3	2	4	1	2	2	1	5

start
 ↓
 end
 ↑

JavaScript 투 포인터 투 포인터 이해하기

특정한 합을 가지는 부분 연속 수열 찾기

JavaScript
투 포인터
투 포인터
이해하기

```
let n = 8; // 데이터의 개수 N
let m = 5; // 찾고자 하는 부분합 M
let data = [3, 2, 4, 1, 2, 2, 1, 5]; // 전체 수열

let cnt = 0;
let intervalSum = 0;
let end = 0;

// start를 차례대로 증가시키며 반복
for (let start = 0; start < n; start++) {
  // end를 가능한 만큼 이동시키기
  while (intervalSum < m && end < n) {
    intervalSum += data[end];
    end += 1;
  }
  // 부분합이 m일 때 카운트 증가
  if (intervalSum == m) cnt += 1;
  intervalSum -= data[start];
}
console.log(cnt);
```