

최장 증가 부분 수열

LIS

(Longest Increasing Subsequence)

부분 수열 (Subsequence)

- 원본 수열의 일부만 뽑아서 원래 순서대로 나열한 수열
- 원본 수열의 일부 원소를 제거한 수열

$$\begin{array}{l} (x_n) := (x_1, x_2, x_3, x_4, x_5, \dots) \\ \quad \quad \quad \vdots \quad \quad \quad \vdots \quad \quad \quad \vdots \\ (x_{n_k}) := (x_1, \quad x_3, \quad x_5, \dots) \end{array}$$

최장 증가 부분 수열 (LIS)

- 부분 수열 중 원소가 오름차순으로 정렬되어 있으면서(= 증가하면서) 가장 긴 수열

A	5	1	6	2	7	3	8
---	---	---	---	---	---	---	---

첫 번째 알고리즘

- 1차열 배열 dp를 정의
- $dp[i]$ = A[i]를 마지막 원소로 가지는 부분 수열 중 LIS의 길이
- $dp[i] = 0 \sim i - 1$ 인 j에 대해서 $A[j] < A[i]$ 인 LIS의 길이($dp[j]$) 중 최대값 + 1
- $$dp[i] = \max_{j=0}^{i-1} \begin{cases} dp[j], & \text{if } A[j] < A[i] \\ 0, & \text{else} \end{cases} + 1$$

첫 번째 알고리즘

A	5	1	6	2	7	3	8
----------	---	---	---	---	---	---	---

dp							
-----------	--	--	--	--	--	--	--

첫 번째 알고리즘

MAX = 0

A							
	5	1	6	2	7	3	8

dp							

첫 번째 알고리즘

$MAX = 0$

A	5	1	6	2	7	3	8

dp	1						

$MAX + 1 = 1$

마지막 원소가 5일때 최대 길이 1

첫 번째 알고리즘

MAX = 0

A	5	1	6	2	7	3	8
----------	---	---	---	---	---	---	---

dp	1						
-----------	---	--	--	--	--	--	--

첫 번째 알고리즘

MAX = 0

5 < 1?

A

5	1	6	2	7	3	8
---	---	---	---	---	---	---

dp

1						
---	--	--	--	--	--	--

첫 번째 알고리즘

$MAX = 0$

A

5	1	6	2	7	3	8
---	---	---	---	---	---	---

dp

1	1					
---	---	--	--	--	--	--

$MAX + 1 = 1$

마지막 원소가 1일때 최대 길이 1

첫 번째 알고리즘

MAX = 0

A							
	5	1	6	2	7	3	8

dp							
	1	1					

첫 번째 알고리즘

MAX = 1

1 < 6?

A

5	1	6	2	7	3	8
---	---	---	---	---	---	---

dp

1	1					
---	---	--	--	--	--	--

첫 번째 알고리즘

MAX = 1

5 < 6?

A

5	1	6	2	7	3	8
---	---	---	---	---	---	---

dp

1	1					
---	---	--	--	--	--	--

첫 번째 알고리즘

$MAX = 1$

A

5	1	6	2	7	3	8
---	---	---	---	---	---	---

dp

1	1	2				
---	---	---	--	--	--	--

$MAX + 1 = 2$

마지막 원소가 6일때 최대 길이 2

첫 번째 알고리즘

MAX = 0

A							
	5	1	6	2	7	3	8

dp							
	1	1	2				

첫 번째 알고리즘

MAX = 0

6 < 2?

A

5	1	6	2	7	3	8
---	---	---	---	---	---	---

dp

1	1	2				
---	---	---	--	--	--	--

첫 번째 알고리즘

MAX = 1

1 < 2?

A

5	1	6	2	7	3	8
---	---	---	---	---	---	---

dp

1	1	2				
---	---	---	--	--	--	--

첫 번째 알고리즘

MAX = 1

5 < 2?

A

5	1	6	2	7	3	8
---	---	---	---	---	---	---

dp

1	1	2				
---	---	---	--	--	--	--

첫 번째 알고리즘

$MAX = 1$

A

5	1	6	2	7	3	8
---	---	---	---	---	---	---

dp

1	1	2	2			
---	---	---	---	--	--	--

$MAX + 1 = 2$

마지막 원소가 2일때 최대 길이 2

첫 번째 알고리즘

MAX = 0

A							
	5	1	6	2	7	3	8

dp							
	1	1	2	2			

첫 번째 알고리즘

MAX = 2

2 < 7?

A

5	1	6	2	7	3	8
---	---	---	---	---	---	---

dp

1	1	2	2			
---	---	---	---	--	--	--

첫 번째 알고리즘

MAX = 2

6 < 7?

A

5	1	6	2	7	3	8
---	---	---	---	---	---	---

dp

1	1	2	2			
---	---	---	---	--	--	--

첫 번째 알고리즘

MAX = 2

1 < 7?

A

5	1	6	2	7	3	8
---	---	---	---	---	---	---

dp

1	1	2	2			
---	---	---	---	--	--	--

첫 번째 알고리즘

MAX = 2

5 < 7?

A

5	1	6	2	7	3	8
---	---	---	---	---	---	---

dp

1	1	2	2			
---	---	---	---	--	--	--

첫 번째 알고리즘

MAX = 2

A	5	1	6	2	7	3	8
----------	---	---	---	---	---	---	---

dp	1	1	2	2	3		
-----------	---	---	---	---	---	--	--

MAX + 1 = 3

마지막 원소가 7일때 최대 길이 3

첫 번째 알고리즘

MAX = 0

A	5	1	6	2	7	3	8
----------	---	---	---	---	---	---	---

dp	1	1	2	2	3		
-----------	---	---	---	---	---	--	--

첫 번째 알고리즘

MAX = 0

7 < 3?

A

5	1	6	2	7	3	8
---	---	---	---	---	---	---

dp

1	1	2	2	3		
---	---	---	---	---	--	--

첫 번째 알고리즘

MAX = 2

2 < 3?

A

5	1	6	2	7	3	8
---	---	---	---	---	---	---

dp

1	1	2	2	3		
---	---	---	---	---	--	--

첫 번째 알고리즘

MAX = 2

6 < 3?

A

5	1	6	2	7	3	8
---	---	---	---	---	---	---

dp

1	1	2	2	3		
---	---	---	---	---	--	--

첫 번째 알고리즘

MAX = 2

1 < 3?

A

5	1	6	2	7	3	8
---	---	---	---	---	---	---

dp

1	1	2	2	3		
---	---	---	---	---	--	--

첫 번째 알고리즘

MAX = 2

5 < 3?

A

5	1	6	2	7	3	8
---	---	---	---	---	---	---

dp

1	1	2	2	3		
---	---	---	---	---	--	--

첫 번째 알고리즘

MAX = 2

A	5	1	6	2	7	3	8
----------	---	---	---	---	---	---	---

dp	1	1	2	2	3	3	
-----------	---	---	---	---	---	---	--

MAX + 1 = 3

마지막 원소가 3일때 최대 길이 3

첫 번째 알고리즘

MAX = 0

A							
	5	1	6	2	7	3	8

dp							
	1	1	2	2	3	3	

첫 번째 알고리즘

MAX = 3

3 < 8?

A

5	1	6	2	7	3	8
---	---	---	---	---	---	---

dp

1	1	2	2	3	3	
---	---	---	---	---	---	--

첫 번째 알고리즘

MAX = 3

7 < 8?

A

5	1	6	2	7	3	8
---	---	---	---	---	---	---

dp

1	1	2	2	3	3	
---	---	---	---	---	---	--

첫 번째 알고리즘

MAX = 3

2 < 8?

A

5	1	6	2	7	3	8
---	---	---	---	---	---	---

dp

1	1	2	2	3	3	
---	---	---	---	---	---	--

첫 번째 알고리즘

MAX = 3

6 < 8?

A

5	1	6	2	7	3	8
---	---	---	---	---	---	---

dp

1	1	2	2	3	3	
---	---	---	---	---	---	--

첫 번째 알고리즘

MAX = 3

1 < 8?

A

5	1	6	2	7	3	8
---	---	---	---	---	---	---

dp

1	1	2	2	3	3	
---	---	---	---	---	---	--

첫 번째 알고리즘

MAX = 3

5 < 8?

A

5	1	6	2	7	3	8
---	---	---	---	---	---	---

dp

1	1	2	2	3	3	
---	---	---	---	---	---	--

첫 번째 알고리즘

MAX = 3

A	5	1	6	2	7	3	8
----------	---	---	---	---	---	---	---

dp	1	1	2	2	3	3	4
-----------	---	---	---	---	---	---	---

MAX + 1 = 4

마지막 원소가 8일때 최대 길이 4

첫 번째 알고리즘

A	5	1	6	2	7	3	8
----------	---	---	---	---	---	---	---

dp	1	1	2	2	3	3	4
-----------	---	---	---	---	---	---	---

첫 번째 알고리즘

A	5	1	6	2	7	3	8
----------	---	---	---	---	---	---	---

dp	1	1	2	2	3	3	4
-----------	---	---	---	---	---	---	---



LIS 길이

첫 번째 알고리즘

시간복잡도 : $O(N^2)$

A	5	1	6	2	7	3	8
----------	---	---	---	---	---	---	---

dp	1	1	2	2	3	3	4
-----------	---	---	---	---	---	---	---



LIS 길이

첫 번째 알고리즘

- 최장 증가 부분 수열 구하기

	0	1	2	3	4	5	6
A	5	1	6	2	7	3	8

dp							
----	--	--	--	--	--	--	--

parent							
--------	--	--	--	--	--	--	--

첫 번째 알고리즘

- 최장 증가 부분 수열 구하기

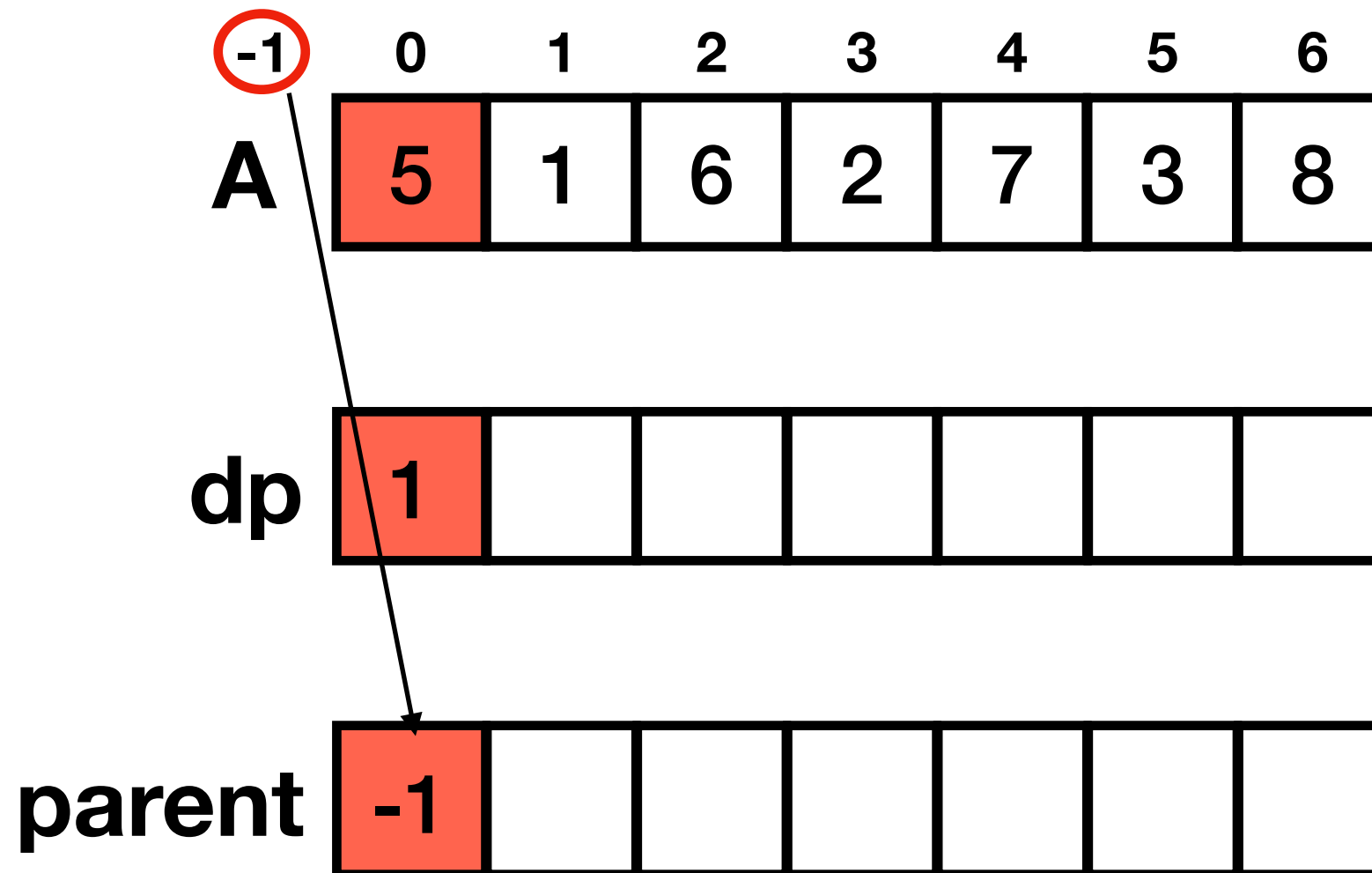
	0	1	2	3	4	5	6
A	5	1	6	2	7	3	8

dp							
----	--	--	--	--	--	--	--

parent							
--------	--	--	--	--	--	--	--

첫 번째 알고리즘

- 최장 증가 부분 수열 구하기



첫 번째 알고리즘

- 최장 증가 부분 수열 구하기

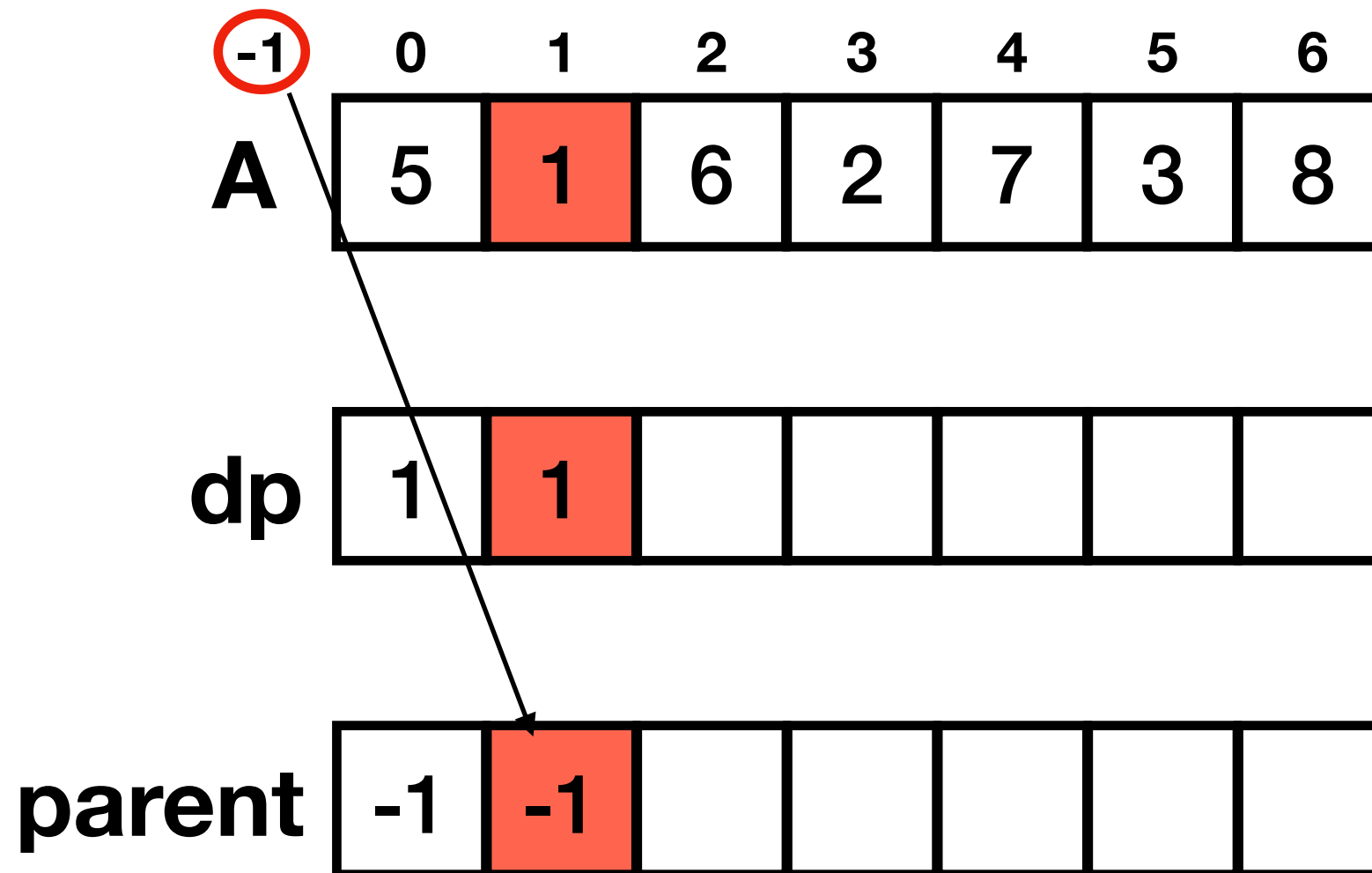
	0	1	2	3	4	5	6
A	5	1	6	2	7	3	8

dp	1						
----	---	--	--	--	--	--	--

parent	-1						
--------	----	--	--	--	--	--	--

첫 번째 알고리즘

- 최장 증가 부분 수열 구하기



첫 번째 알고리즘

- 최장 증가 부분 수열 구하기

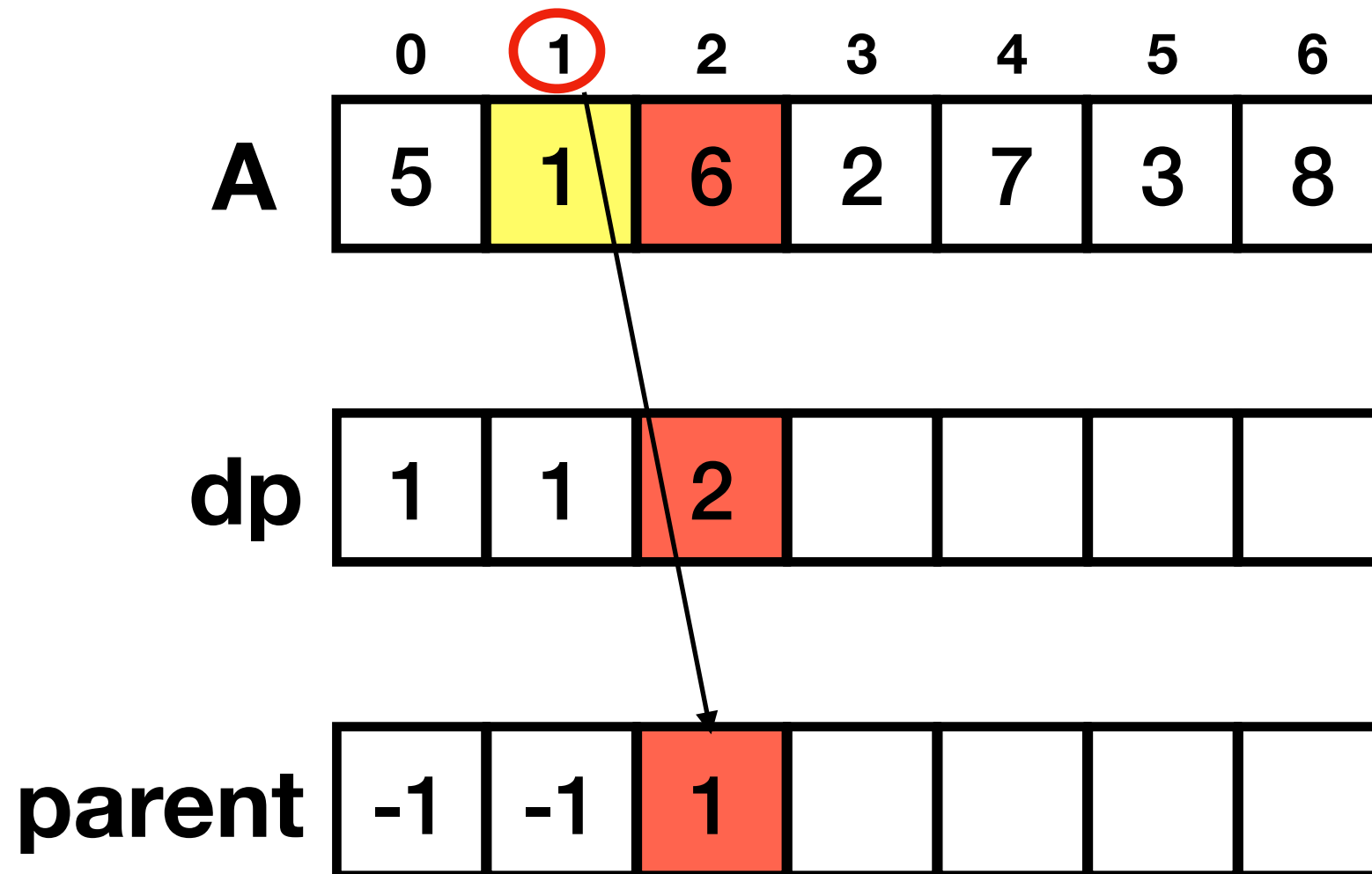
	0	1	2	3	4	5	6
A	5	1	6	2	7	3	8

dp	1	1					
----	---	---	--	--	--	--	--

parent	-1	-1					
--------	----	----	--	--	--	--	--

첫 번째 알고리즘

- 최장 증가 부분 수열 구하기



첫 번째 알고리즘

- 최장 증가 부분 수열 구하기

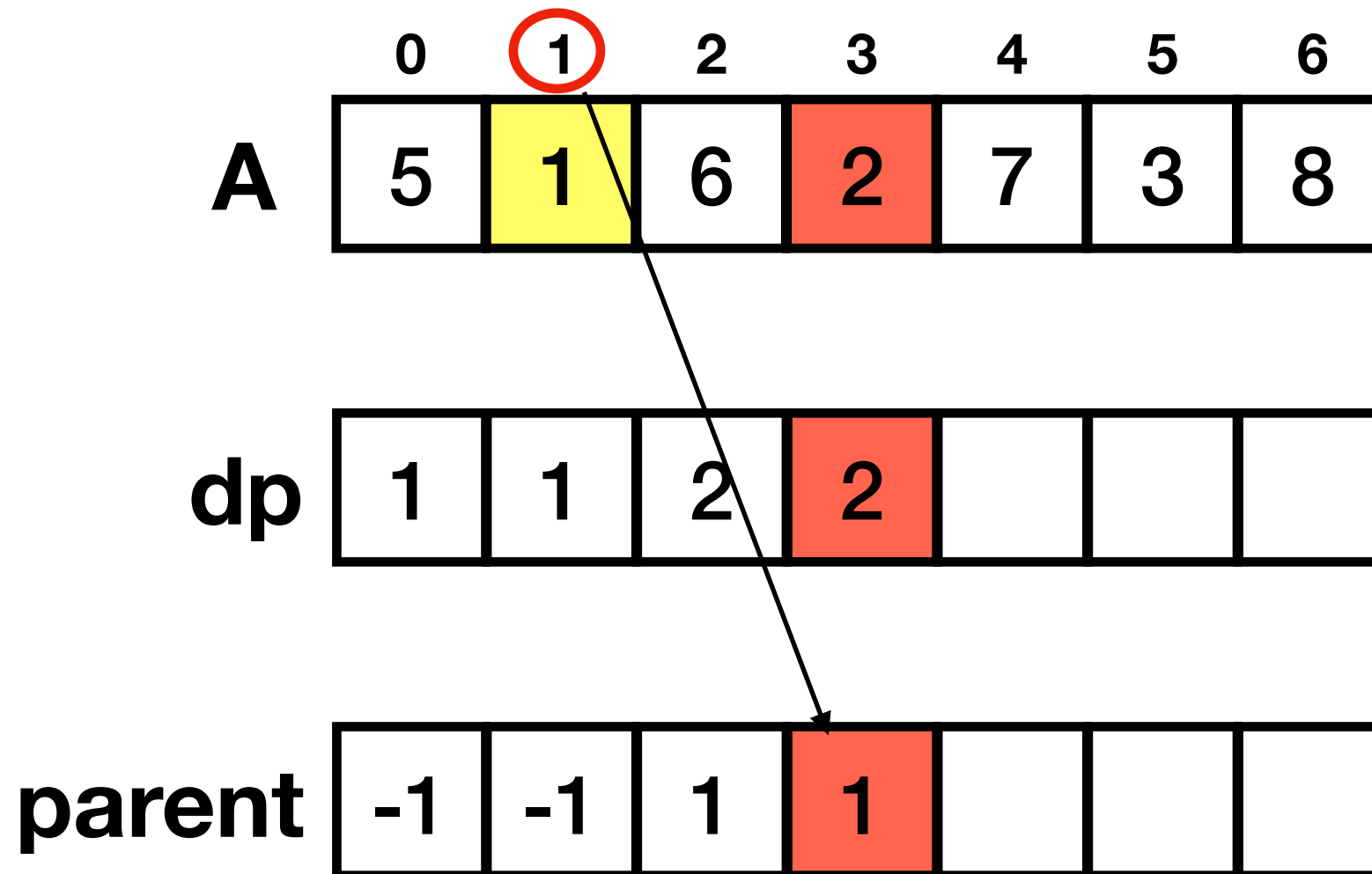
	0	1	2	3	4	5	6
A	5	1	6	2	7	3	8

dp	1	1	2				
----	---	---	---	--	--	--	--

parent	-1	-1	1				
--------	----	----	---	--	--	--	--

첫 번째 알고리즘

- 최장 증가 부분 수열 구하기



첫 번째 알고리즘

- 최장 증가 부분 수열 구하기

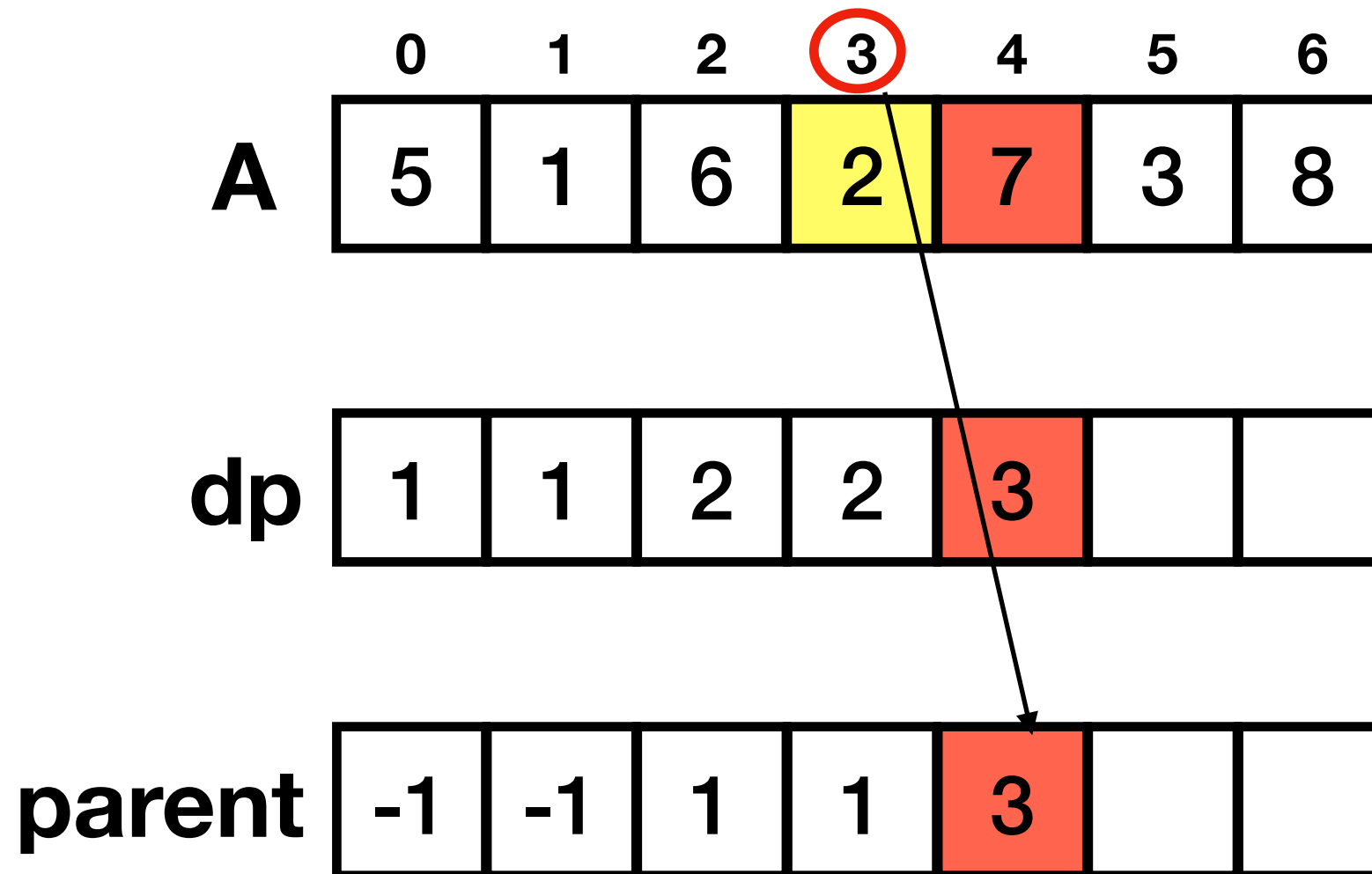
	0	1	2	3	4	5	6
A	5	1	6	2	7	3	8

dp	1	1	2	2			
----	---	---	---	---	--	--	--

parent	-1	-1	1	1			
--------	----	----	---	---	--	--	--

첫 번째 알고리즘

- 최장 증가 부분 수열 구하기



첫 번째 알고리즘

- 최장 증가 부분 수열 구하기

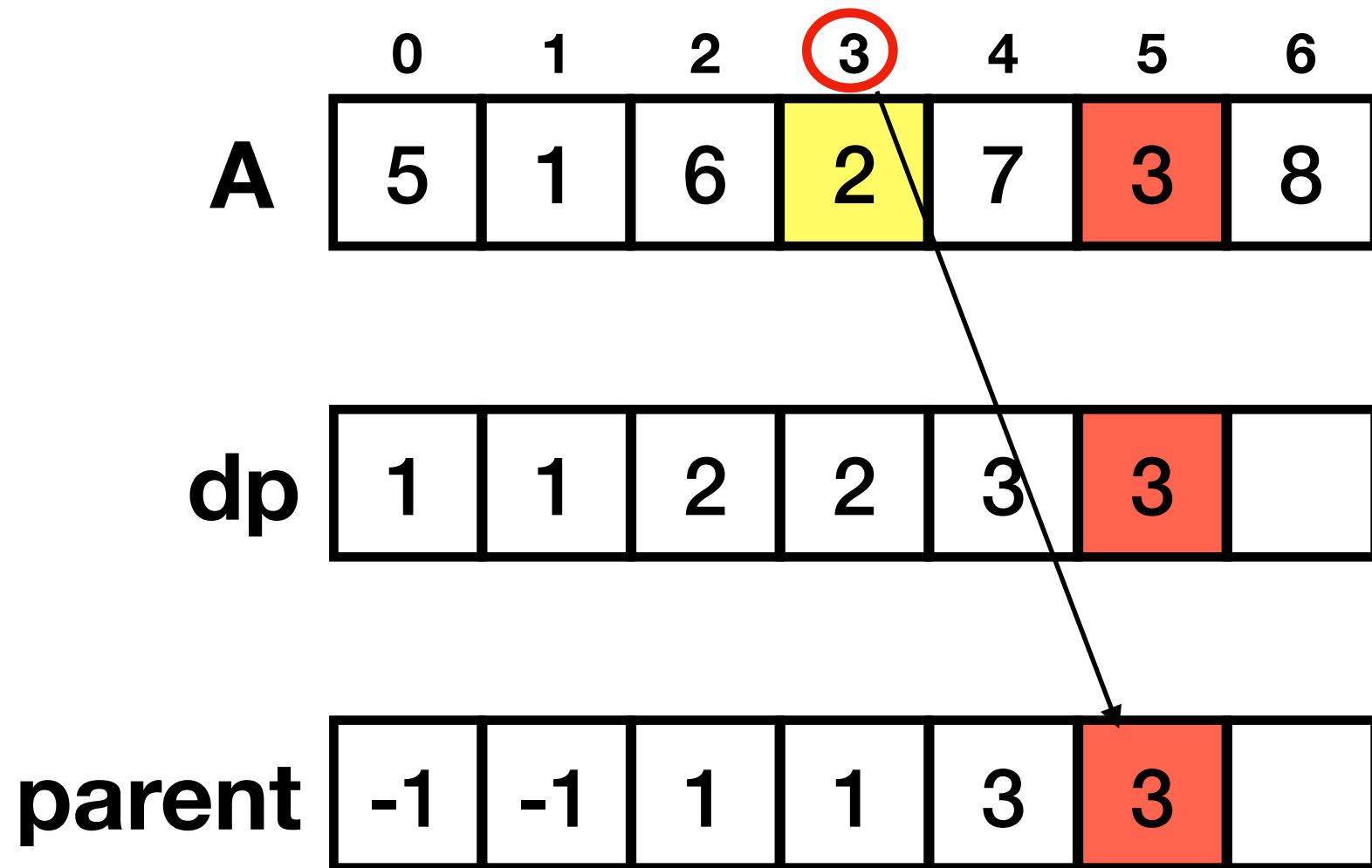
	0	1	2	3	4	5	6
A	5	1	6	2	7	3	8

dp	1	1	2	2	3		
----	---	---	---	---	---	--	--

parent	-1	-1	1	1	3		
--------	----	----	---	---	---	--	--

첫 번째 알고리즘

- 최장 증가 부분 수열 구하기



첫 번째 알고리즘

- 최장 증가 부분 수열 구하기

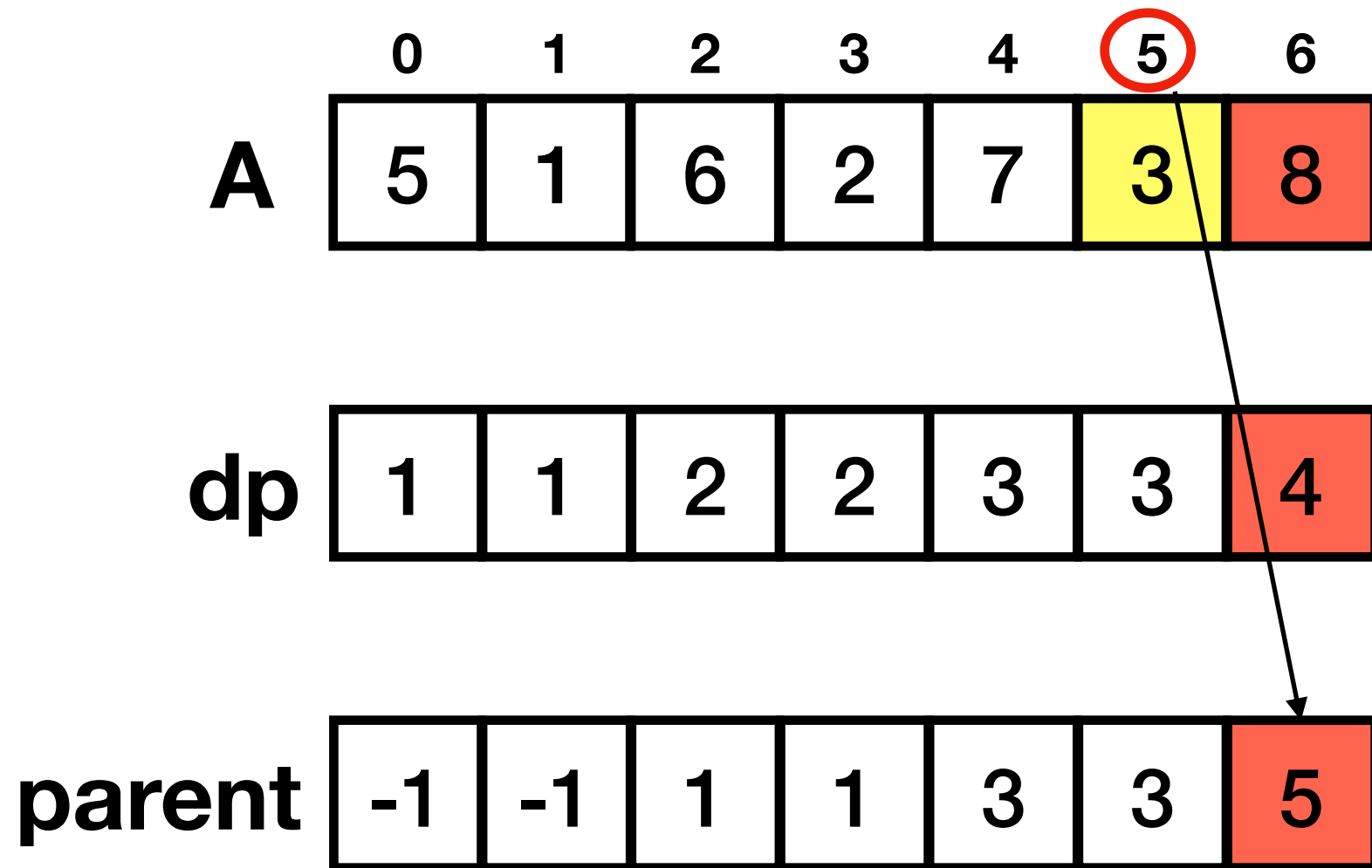
	0	1	2	3	4	5	6
A	5	1	6	2	7	3	8

dp	1	1	2	2	3	3	
-----------	---	---	---	---	---	---	--

parent	-1	-1	1	1	3	3	
---------------	----	----	---	---	---	---	--

첫 번째 알고리즘

- 최장 증가 부분 수열 구하기



첫 번째 알고리즘

- 최장 증가 부분 수열 구하기

A

5	1	6	2	7	3	8
---	---	---	---	---	---	---

LIS 마지막 원소

dp

1	1	2	2	3	3	4
---	---	---	---	---	---	---

LIS 길이

첫 번째 알고리즘

- 최장 증가 부분 수열 구하기

	0	1	2	3	4	5	6
A	5	1	6	2	7	3	8

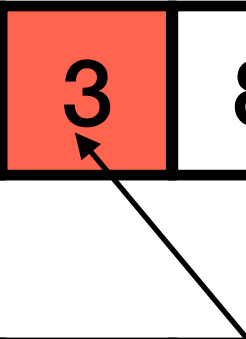
parent	-1	-1	1	1	3	3	5
---------------	----	----	---	---	---	---	---

8

첫 번째 알고리즘

- 최장 증가 부분 수열 구하기

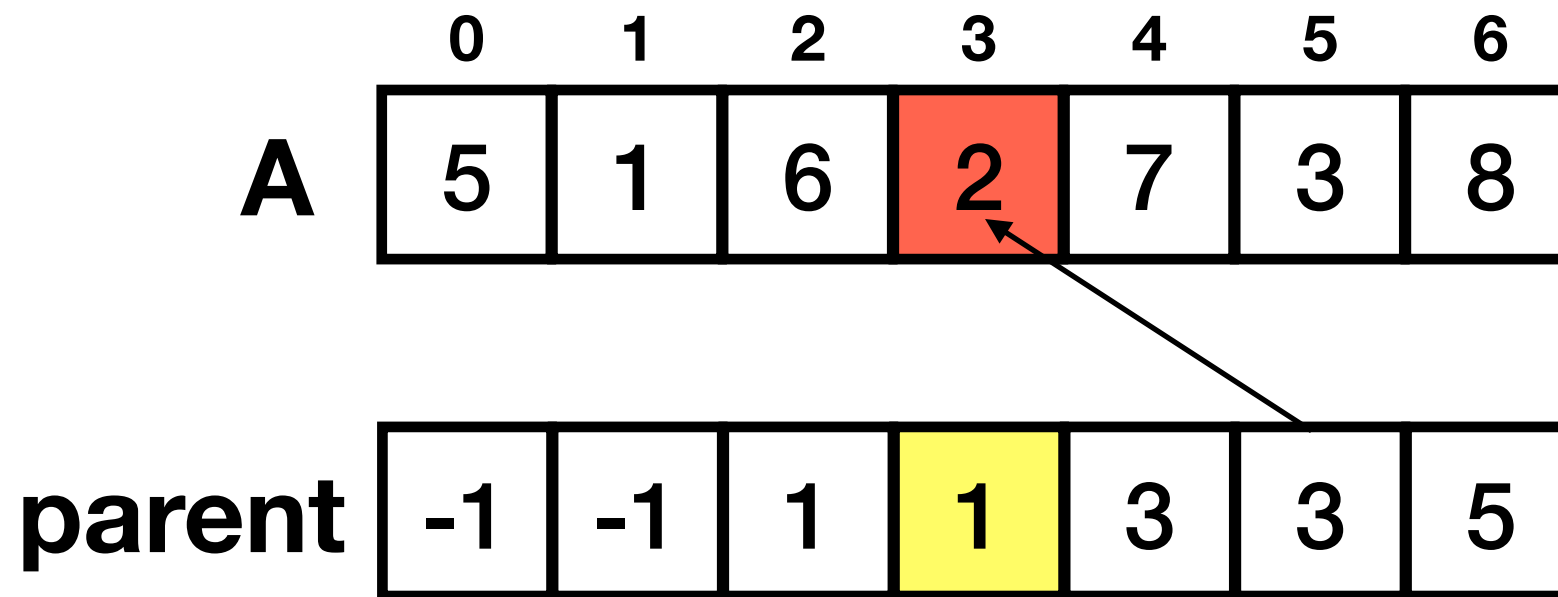
	0	1	2	3	4	5	6
A	5	1	6	2	7	3	8
parent	-1	-1	1	1	3	3	5



3 -> 8

첫 번째 알고리즘

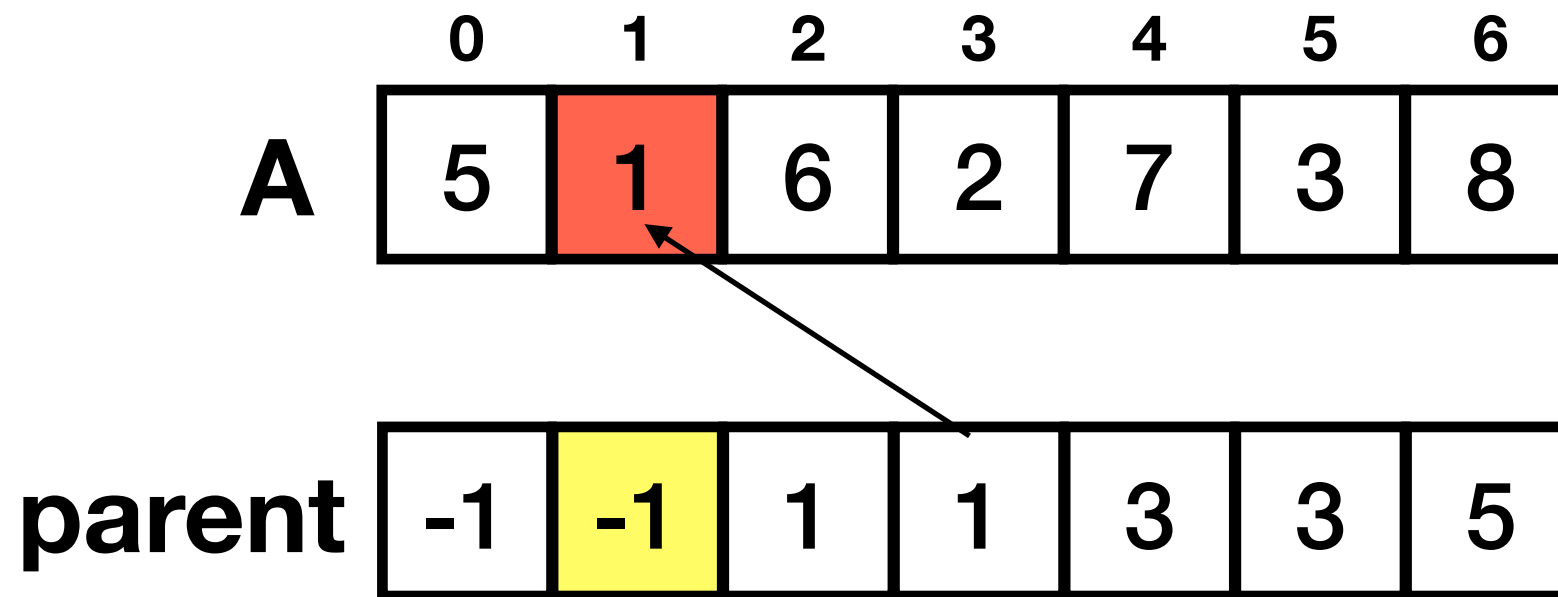
- 최장 증가 부분 수열 구하기



2 -> 3 -> 8

첫 번째 알고리즘

- 최장 증가 부분 수열 구하기



1 -> 2 -> 3 -> 8

첫 번째 알고리즘

- 최장 증가 부분 수열 구하기

	0	1	2	3	4	5	6
A	5	1	6	2	7	3	8

parent	-1	-1	1	1	3	3	5
---------------	----	----	---	---	---	---	---

LIS : 1 -> 2 -> 3 -> 8

알고리즘 개선 아이디어

- $dp[i]$ 를 계산하기 위해서 $dp[0] \sim dp[i-1]$ 다 훑어봐야했던 이유
-> $A[i]$ 보다 작은 원소이면서 가장 큰 $dp[j]$ 를 찾기 위함
- $dp[i]$ 가 같은 요소들끼리 묶고 그 중에서 $A[i]$ 가 최소인 값을 대표로 저장하고 있으면 위치를 빠르게 알 수 있다.
- 마지막 원소가 $A[i]$ 일때 LIS의 길이를 $dp[i]$ 에 저장
(변경)-> **LIS의 길이가 i 일때 마지막 원소로 올 수 있는 값 중 최소 값을 $dp[i]$ 에 저장**

알고리즘 개선 아이디어

A

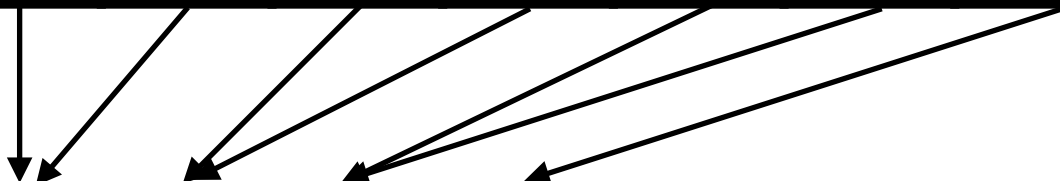
5	1	6	2	7	3	8
---	---	---	---	---	---	---

dp

1	1	2	2	3	3	4
---	---	---	---	---	---	---

dp2

1	2	3	8
---	---	---	---



두 번째 알고리즘

- 1차열 배열 dp 를 정의
- $dp[i]$ = 길이가 i 인 LIS의 마지막 원소들 중 최소값
- **$A[i]$ 가 포함된 LIS의 길이(k)** = $A[i]$ 보다 작은 $dp[i]$ 중 가장 큰 값의 최대 길이(=인덱스, i) + 1
- $dp[k] = A[i]$ (길이가 k 인 LIS의 마지막 원소 중 최소)

두 번째 알고리즘

A	5	1	6	2	7	3	8
---	---	---	---	---	---	---	---

dp							
----	--	--	--	--	--	--	--

두 번째 알고리즘

A	5	1	6	2	7	3	8
---	---	---	---	---	---	---	---

dp	5						
----	---	--	--	--	--	--	--

LIS의 길이가 1일때 마지막 원소의 최소값 5

두 번째 알고리즘

A	5	1	6	2	7	3	8
----------	---	---	---	---	---	---	---

dp	5						
-----------	---	--	--	--	--	--	--

두 번째 알고리즘

A

5	1	6	2	7	3	8
---	---	---	---	---	---	---

5 < 1?

dp

5						
---	--	--	--	--	--	--

두 번째 알고리즘

A	5	1	6	2	7	3	8
---	---	---	---	---	---	---	---

dp	1						
----	---	--	--	--	--	--	--

LIS의 길이가 1일때 마지막 원소의 최소값 1

두 번째 알고리즘

A							
	5	1	6	2	7	3	8

dp							
	1						

두 번째 알고리즘

A

5	1	6	2	7	3	8
---	---	---	---	---	---	---

dp

1 < 6?

1						
---	--	--	--	--	--	--

두 번째 알고리즘

A	5	1	6	2	7	3	8
---	---	---	---	---	---	---	---

dp	1	6					
----	---	---	--	--	--	--	--

LIS의 길이가 2일때 마지막 원소의 최소값 6

두 번째 알고리즘

A							
	5	1	6	2	7	3	8

dp							
	1	6					

두 번째 알고리즘

A

5	1	6	2	7	3	8
---	---	---	---	---	---	---

dp

1	6					
---	---	--	--	--	--	--

6 < 2?

두 번째 알고리즘

A

5	1	6	2	7	3	8
---	---	---	---	---	---	---

dp

1<2?

1	6					
---	---	--	--	--	--	--

두 번째 알고리즘

A							
	5	1	6	2	7	3	8

dp							
	1	2					

LIS의 길이가 2일때 마지막 원소의 최소값 2

두 번째 알고리즘

A	5	1	6	2	7	3	8
----------	---	---	---	---	---	---	---

dp	1	2					
-----------	---	---	--	--	--	--	--

두 번째 알고리즘

A

5	1	6	2	7	3	8
---	---	---	---	---	---	---

$2 < 7?$

dp

1	2					
---	---	--	--	--	--	--

두 번째 알고리즘

A	5	1	6	2	7	3	8
---	---	---	---	---	---	---	---

dp	1	2	7				
----	---	---	---	--	--	--	--

LIS의 길이가 3일때 마지막 원소의 최소값 7

두 번째 알고리즘

A	5	1	6	2	7	3	8
----------	---	---	---	---	---	---	---

dp	1	2	7				
-----------	---	---	---	--	--	--	--

두 번째 알고리즘

A

5	1	6	2	7	3	8
---	---	---	---	---	---	---

7<3?

dp

1	2	7				
---	---	---	--	--	--	--

두 번째 알고리즘

A

5	1	6	2	7	3	8
---	---	---	---	---	---	---

dp

2<3?

1	2	7				
---	---	---	--	--	--	--

두 번째 알고리즘

A							
	5	1	6	2	7	3	8

dp							
	1	2	3				

LIS의 길이가 3일때 마지막 원소의 최소값 3

두 번째 알고리즘

A	5	1	6	2	7	3	8
----------	---	---	---	---	---	---	---

dp	1	2	3				
-----------	---	---	---	--	--	--	--

두 번째 알고리즘

A

5	1	6	2	7	3	8
---	---	---	---	---	---	---

3<8?

dp

1	2	3				
---	---	---	--	--	--	--

두 번째 알고리즘

A	5	1	6	2	7	3	8
----------	---	---	---	---	---	---	---

dp	1	2	3	8			
-----------	---	---	---	---	--	--	--

LIS의 길이가 4일때 마지막 원소의 최소값 8

두 번째 알고리즘

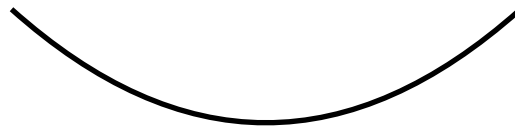
시간복잡도 : $O(N^2)$?

A

5	1	6	2	7	3	8
---	---	---	---	---	---	---

dp

1	2	3	8			
---	---	---	---	--	--	--



dp 배열 크기 = LIS 길이 = 4

알고리즘 개선 아이디어

dp	1	2	3	8			
----	---	---	---	---	--	--	--

- A[i]보다 작은 값 중 가장 큰 값의 다음 위치에 A[i] 넣기
(= A[i] 이상은 값 중 가장 작은 값의 위치에 A[i] 넣기)
- dp 배열은 항상 오름차순 정렬 상태

이분탐색 (LowerBound)

$$O(N^2) \rightarrow O(N * \text{Log}N)$$

두 번째 알고리즘

- 최장 증가 부분 수열 구하기

	0	1	2	3	4	5	6
A	5	1	6	2	7	3	8

dp							
-----------	--	--	--	--	--	--	--

parent							
---------------	--	--	--	--	--	--	--

두 번째 알고리즘

- 최장 증가 부분 수열 구하기

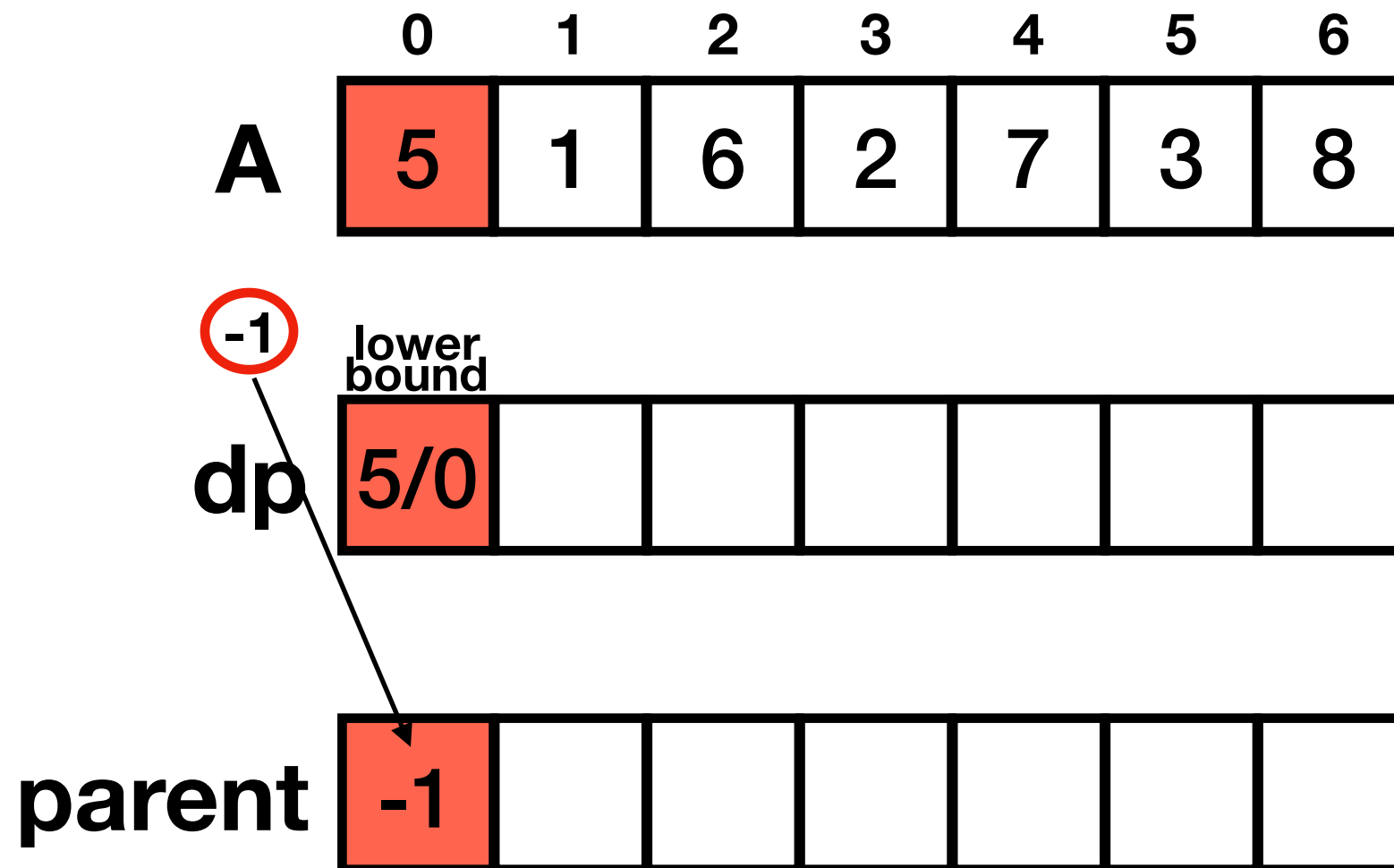
	0	1	2	3	4	5	6
A	5	1	6	2	7	3	8

dp							
----	--	--	--	--	--	--	--

parent							
--------	--	--	--	--	--	--	--

두 번째 알고리즘

- 최장 증가 부분 수열 구하기



두 번째 알고리즘

- 최장 증가 부분 수열 구하기

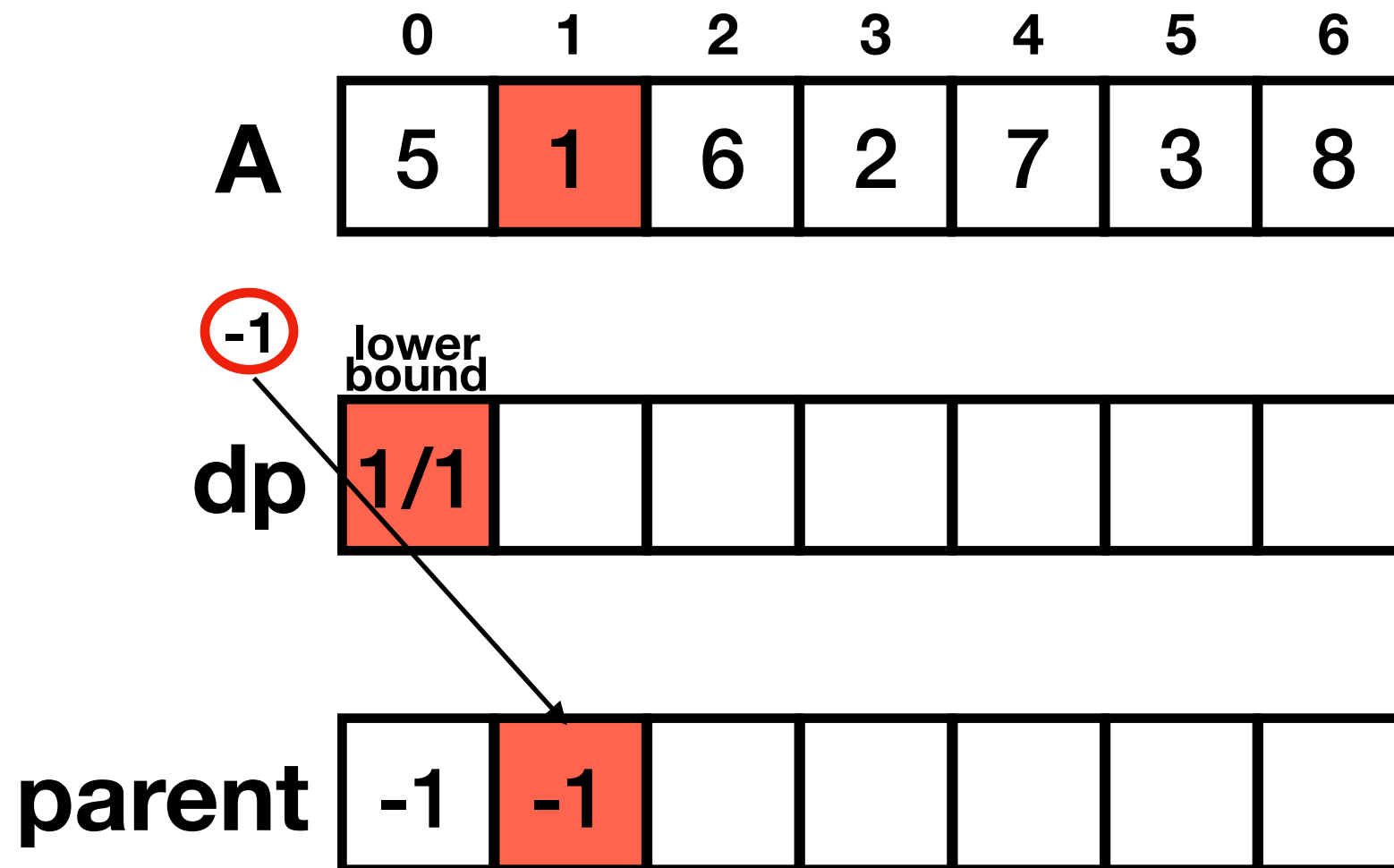
	0	1	2	3	4	5	6
A	5	1	6	2	7	3	8

dp	5/0						
----	-----	--	--	--	--	--	--

parent	-1						
--------	----	--	--	--	--	--	--

두 번째 알고리즘

- 최장 증가 부분 수열 구하기



두 번째 알고리즘

- 최장 증가 부분 수열 구하기

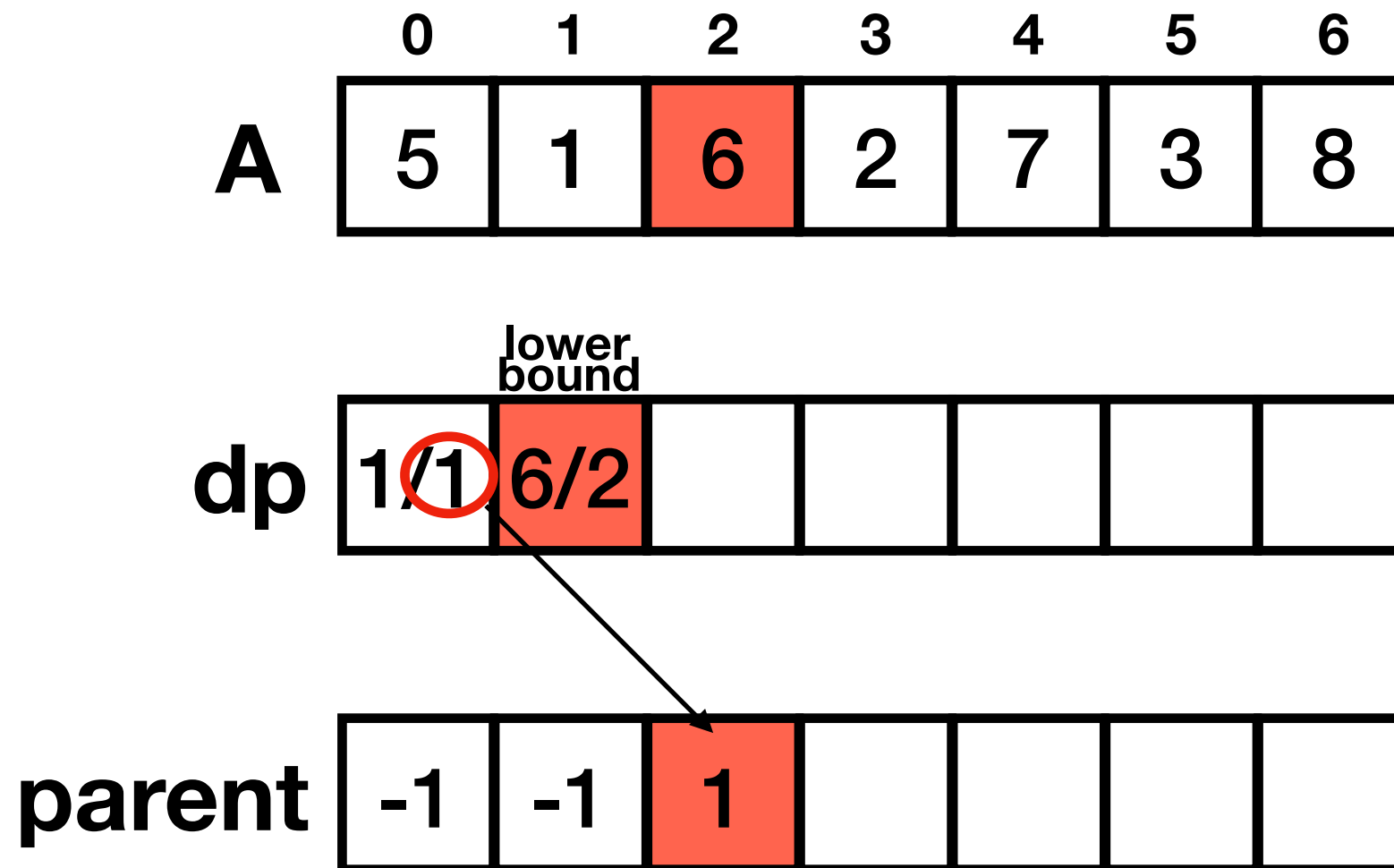
	0	1	2	3	4	5	6
A	5	1	6	2	7	3	8

dp	1/1						
----	-----	--	--	--	--	--	--

parent	-1	-1					
--------	----	----	--	--	--	--	--

두 번째 알고리즘

- 최장 증가 부분 수열 구하기



두 번째 알고리즘

- 최장 증가 부분 수열 구하기

	0	1	2	3	4	5	6
A	5	1	6	2	7	3	8

dp	1/1	6/2					
-----------	-----	-----	--	--	--	--	--

parent	-1	-1	1				
---------------	----	----	---	--	--	--	--

두 번째 알고리즘

- 최장 증가 부분 수열 구하기

	0	1	2	3	4	5	6
A	5	1	6	2	7	3	8

	lower bound					
dp	1/1	2/3				

parent	-1	-1	1	1		
---------------	----	----	---	---	--	--

두 번째 알고리즘

- 최장 증가 부분 수열 구하기

	0	1	2	3	4	5	6
A	5	1	6	2	7	3	8

dp	1/1	2/3					
----	-----	-----	--	--	--	--	--

parent	-1	-1	1	1			
--------	----	----	---	---	--	--	--

두 번째 알고리즘

- 최장 증가 부분 수열 구하기

	0	1	2	3	4	5	6
A	5	1	6	2	7	3	8
			lower bound				
dp	1/1	2/3	7/4				
parent	-1	-1	1	1	3		

두 번째 알고리즘

- 최장 증가 부분 수열 구하기

	0	1	2	3	4	5	6
A	5	1	6	2	7	3	8

dp	1/1	2/3	7/4				
-----------	-----	-----	-----	--	--	--	--

parent	-1	-1	1	1	3		
---------------	----	----	---	---	---	--	--

두 번째 알고리즘

- 최장 증가 부분 수열 구하기

	0	1	2	3	4	5	6
A	5	1	6	2	7	3	8

			lower bound				
dp	1/1	2/2	3/5				

parent	-1	-1	1	1	3	3	
---------------	----	----	---	---	---	---	--

두 번째 알고리즘

- 최장 증가 부분 수열 구하기

	0	1	2	3	4	5	6
A	5	1	6	2	7	3	8

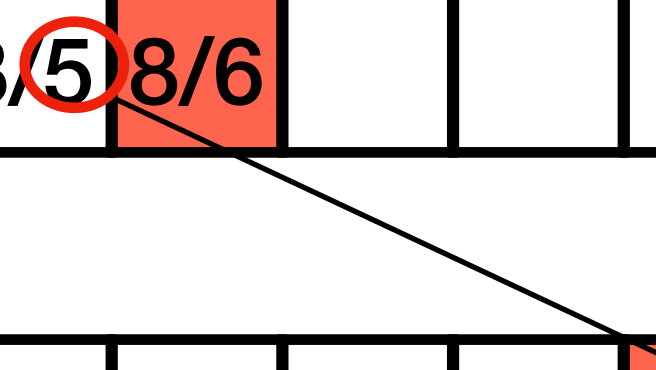
dp	1/1	2/3	3/5				
----	-----	-----	-----	--	--	--	--

parent	-1	-1	1	1	3	3	
--------	----	----	---	---	---	---	--

두 번째 알고리즘

- 최장 증가 부분 수열 구하기

	0	1	2	3	4	5	6
A	5	1	6	2	7	3	8
				lower bound			
dp	1/1	2/3	3/5	8/6			
parent	-1	-1	1	1	3	3	5



두 번째 알고리즘

- 최장 증가 부분 수열 구하기

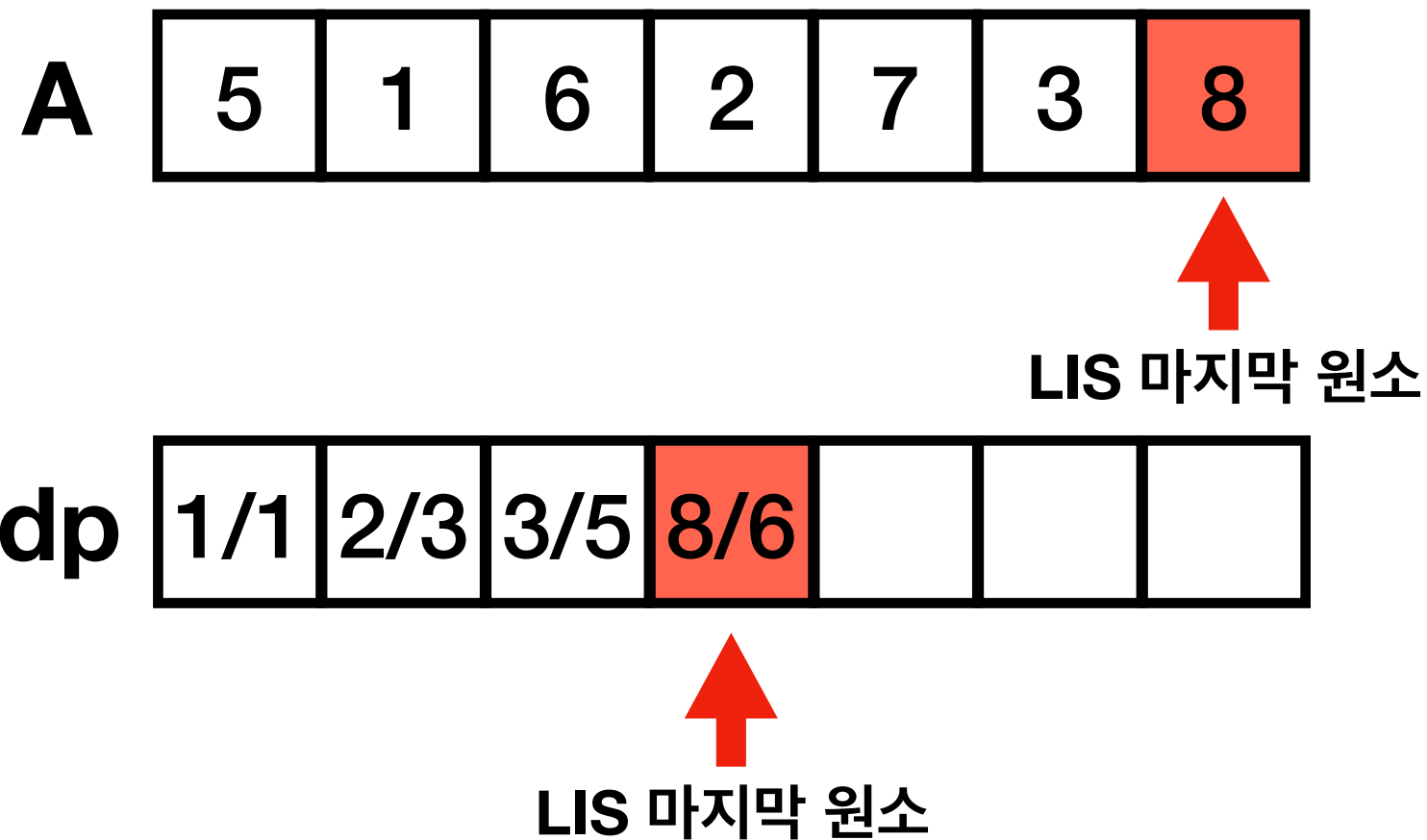
	0	1	2	3	4	5	6
A	5	1	6	2	7	3	8

dp	1/1	2/3	3/5	8/6			
-----------	-----	-----	-----	-----	--	--	--

parent	-1	-1	1	1	3	3	5
---------------	----	----	---	---	---	---	---

첫 번째 알고리즘

- 최장 증가 부분 수열 구하기



첫 번째 알고리즘

- 최장 증가 부분 수열 구하기

	0	1	2	3	4	5	6
A	5	1	6	2	7	3	8

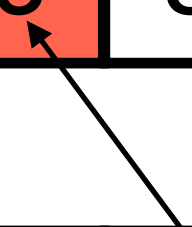
parent	-1	-1	1	1	3	3	5
---------------	----	----	---	---	---	---	---

8

첫 번째 알고리즘

- 최장 증가 부분 수열 구하기

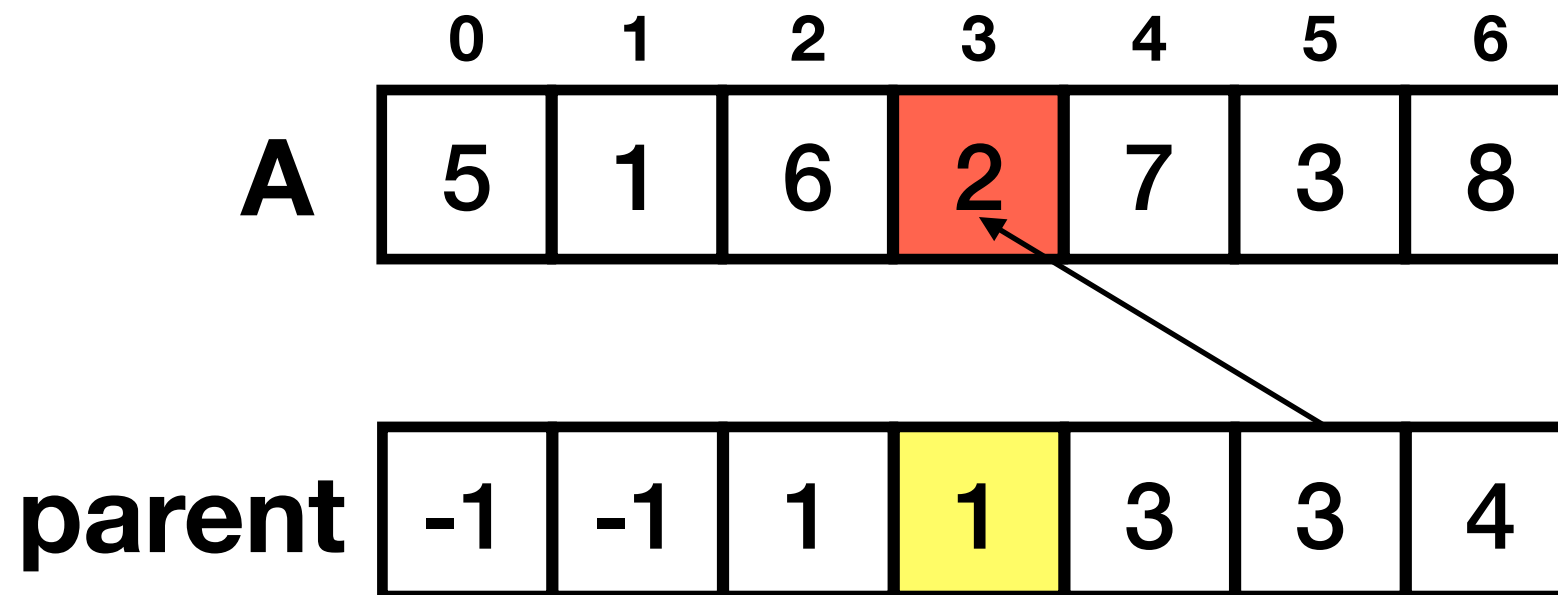
	0	1	2	3	4	5	6
A	5	1	6	2	7	3	8
parent	-1	-1	1	1	3	3	4



3 -> 8

첫 번째 알고리즘

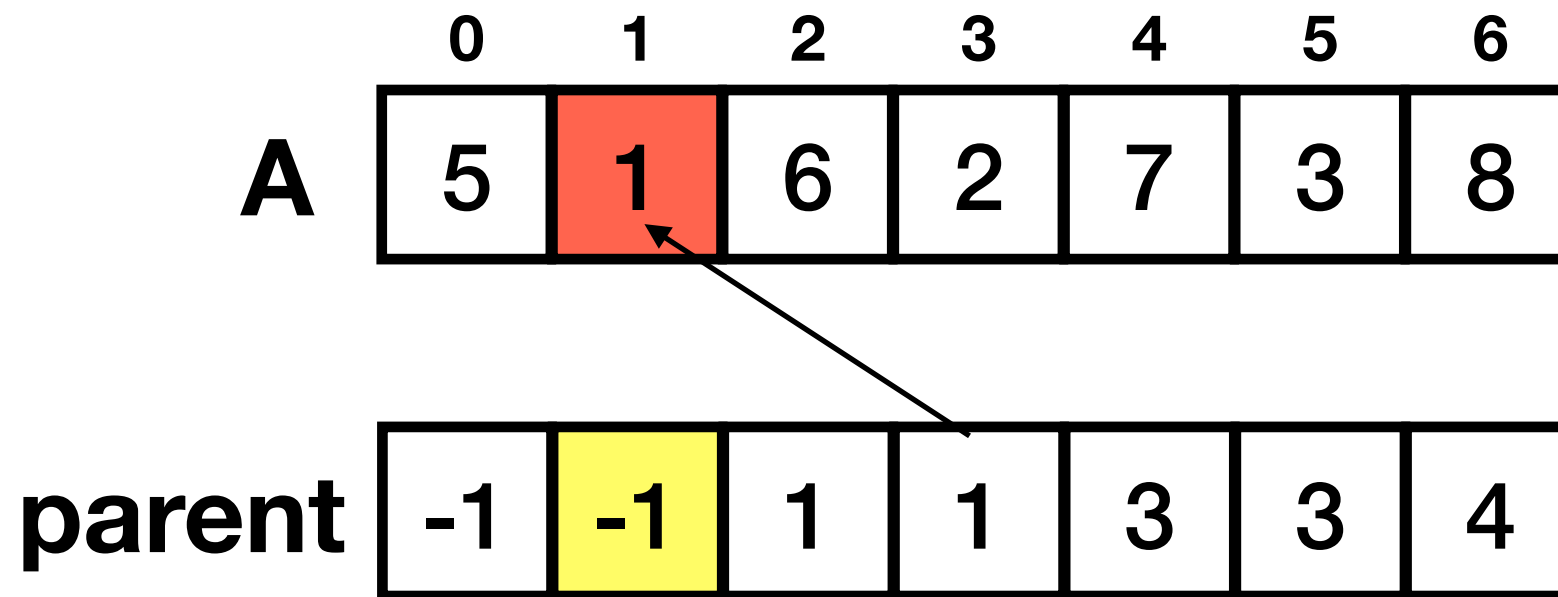
- 최장 증가 부분 수열 구하기



2 -> 3 -> 8

첫 번째 알고리즘

- 최장 증가 부분 수열 구하기



1 -> 2 -> 3 -> 8

첫 번째 알고리즘

- 최장 증가 부분 수열 구하기

	0	1	2	3	4	5	6
A	5	1	6	2	7	3	8

parent	-1	-1	1	1	3	3	4
---------------	----	----	---	---	---	---	---

LIS : 1 -> 2 -> 3 -> 8