

문자열 알고리즘 2

최백준 choi@startlink.io

Suffix Array

Prefix

Prefix

- baekjoon의 Prefix

1. b

2. ba

3. bae

4. baek

5. baekj

6. baekjo

7. baekjoo

8. baekjoon

Suffix

Suffix

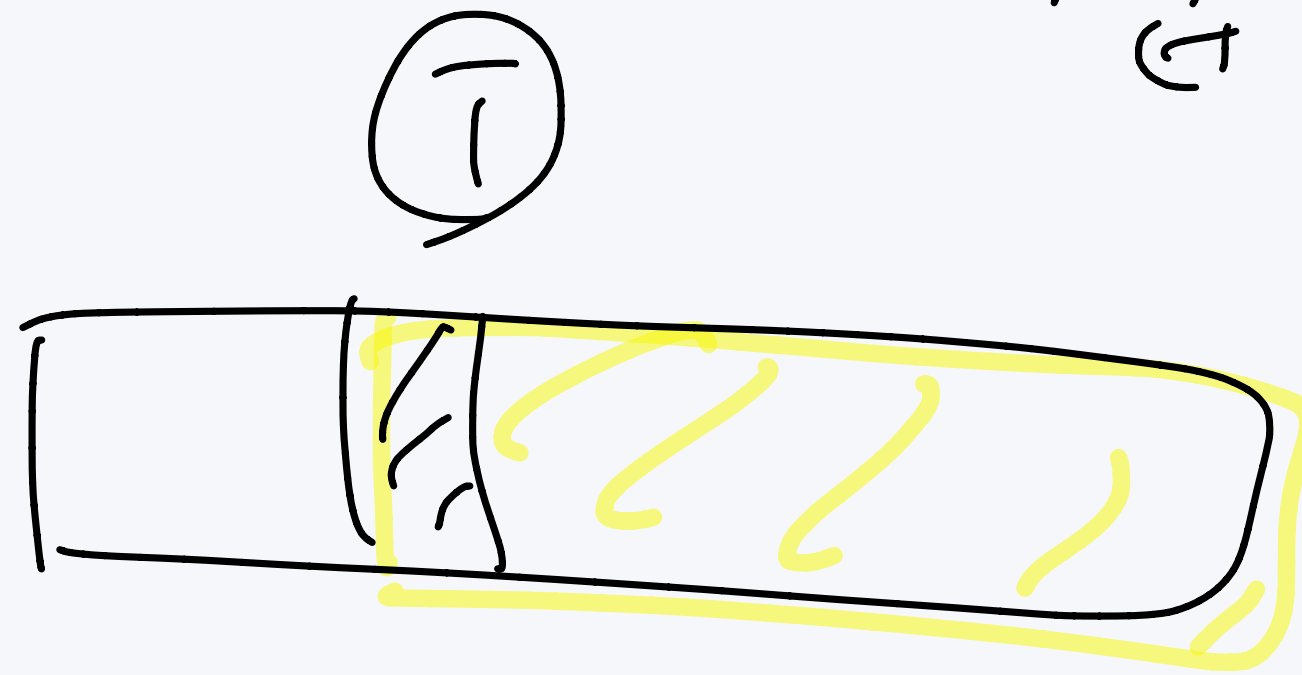
0

- baekjoon의 Suffix

1. baekjoon	0
2. aekjoon	1
3. ekjoon	2
4. kjoon	3
5. joon	4
6. oon	5
7. on	6
8. n	7

가장자리 N인 문자열

가장자리 : N개



↓

가장자리

↓

가장자리

$N^2 \lg N$

접미사 배열

Suffix Array

5

- 문자열 S 의 모든 접미사를 구해서 사전순으로 정렬한 자료구조

접미사 배열

Suffix Array

- aekjoon
- baekjoon
- ekjoon
- joon
- kjoon
- n
- on
- oon

[0 1 2 3 4 5 6]

↓ 정렬

[1 0 2 4 3 5 6]

접미사

Suffix

- 접미사는 정수로 나타낼 수 있다.
- i 번 접미사: i 번째 글자에서 시작하는 접미사

접미사 배열

<https://www.acmicpc.net/problem/11656>

- 문자열 S의 모든 접미사를 구하는 문제

점미사 배열

<https://www.acmicpc.net/problem/11656>

- 정렬할 때, 매번 부분 문자열을 만들지 않기 위해 strcmp를 사용했다

```
vector<int> a(n);  
for (int i=0; i<n; i++) {  
    a[i] = i;  
}  
sort(a.begin(), a.end(), [&s](int u, int v) {  
    return strcmp(s.c_str()+u, s.c_str()+v) < 0;  
});
```

접미사 배열

<https://www.acmicpc.net/problem/11656>

- 소스: <http://boj.kr/339caa50a28b4da2a4d1e69e73090890>

S 의 접미사들의 접미사
||
 S 의 접미사

접미사 배열

<https://www.acmicpc.net/problem/11656>

- 길이가 N인 문자열 S의 접미사는 총 N개가 있다.
- 따라서, N개를 정렬하는데 걸리는 시간은 $O(N \lg N)$ 이다?

S_k ($1 \leq k \leq N$)

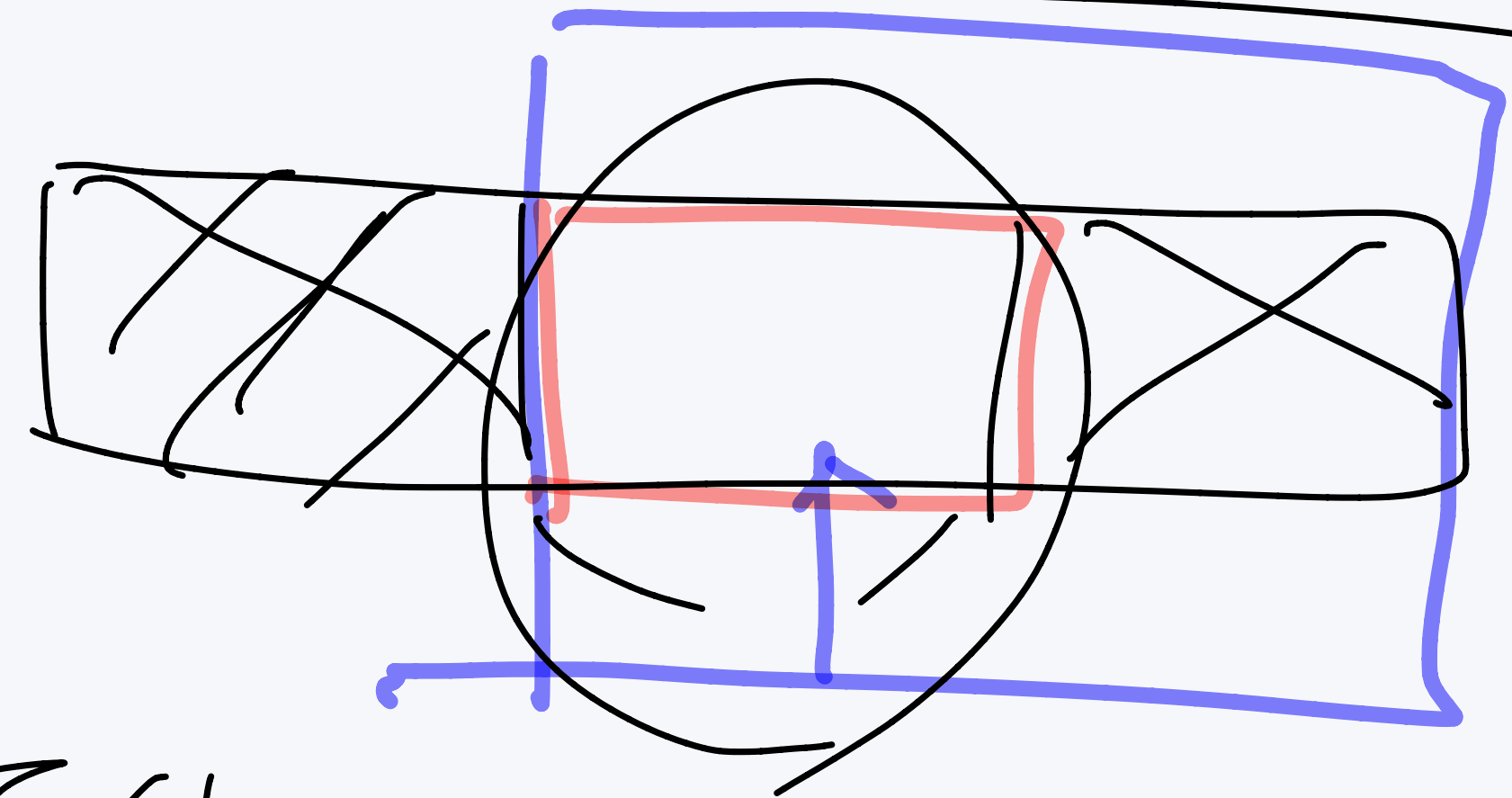
접미사의 정렬 시 -
suffix의 prefix

①

문자열

$\rightarrow [S[1 \dots N]]$

$\rightarrow S$ 의 앞, 뒤 일부 삭제



접미사 배열

<https://www.acmicpc.net/problem/11656>

- 길이가 N 인 문자열 S 의 접미사는 총 N 개가 있다.
- 따라서, N 개를 정렬하는데 걸리는 시간은 $O(N \lg N)$ 이다?
- 아니다. 문자열을 비교하는데 걸리는 시간이 $O(N)$ 이기 때문에, $O(N^2 \lg N)$ 이다.

접미사 배열 \rightarrow 문자열

$S < t$

$S_0 < t_0$

$S_0 = t_0$

$S_1 < t_1$

접미사 배열 2

<https://www.acmicpc.net/problem/13013>

13

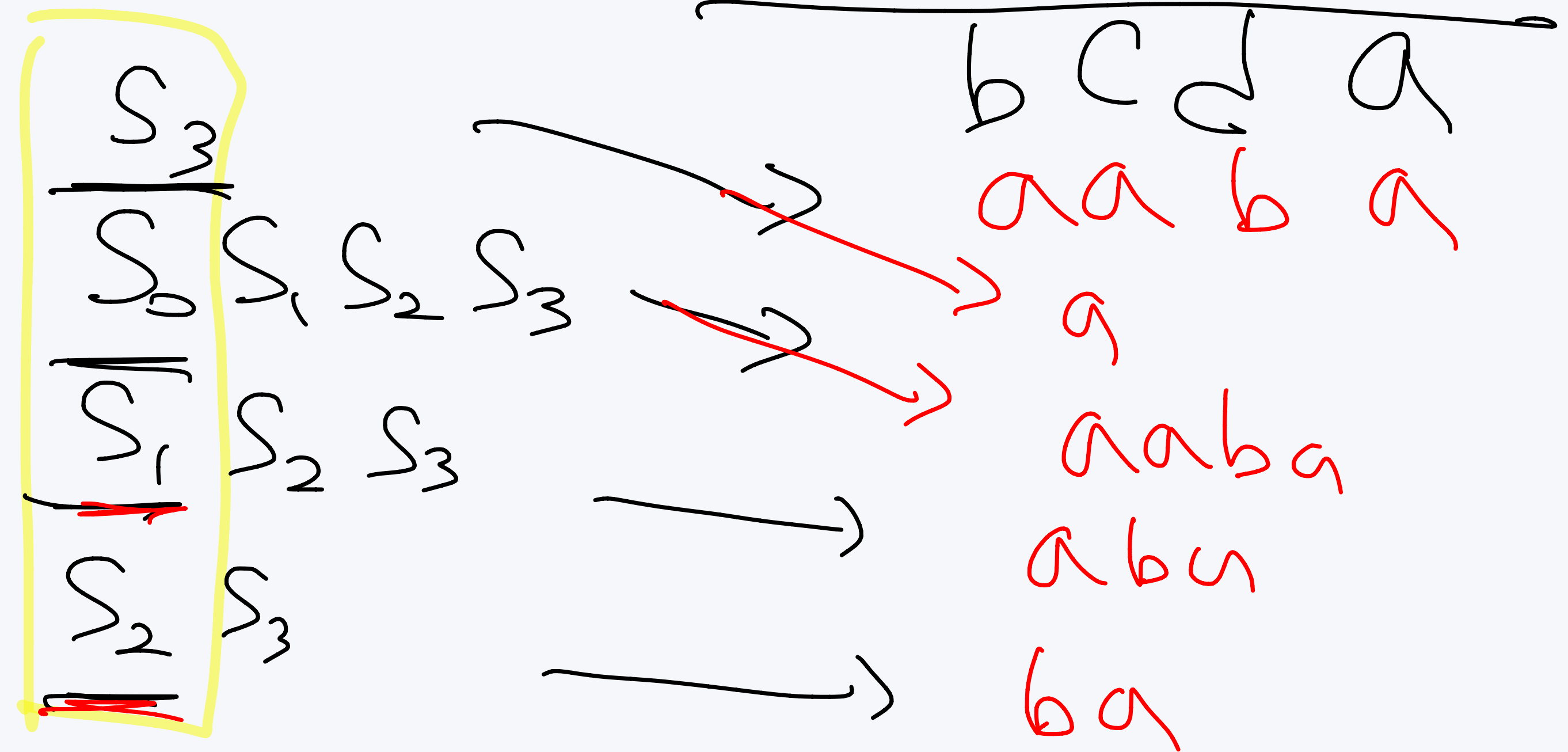
$$SA = [3, 0, 1, 2]$$

$$S = bcda$$

- 접미사 배열 SA가 주어졌을 때, 그러한 접미사 배열을 만드는 문자열 S를 만드는 문제
- 이 때, S에 포함된 서로 다른 문자의 개수가 최소가 되어야 한다

$$S = S_0 S_1 S_2 S_3$$

b			
c	1	2	3
d	2	3	
a	3		



접미사 배열 2

14

<https://www.acmicpc.net/problem/13013>

- 접미사 배열 SA가 주어졌을 때, 그러한 접미사 배열을 만드는 문자열 S를 아무거나 만들어 보자

접미사 배열 2

<https://www.acmicpc.net/problem/13013>

- $SA[i]$ 번째에서 시작하는 접미사는 $SA[i+1]$ 번째에 시작하는 접미사보다 사전순으로 앞서야 한다
- $SA[i]$ 번째 접미사와 $SA[i+1]$ 번째 접미사를 비교한다고 하면
- 먼저, $S[SA[i]]$ 와 $S[SA[i+1]]$ 을 비교해야 한다
- $S[SA[i]] > S[SA[i+1]]$ 은 절대로 일어날 수 없다

접미사 배열 2

<https://www.acmicpc.net/problem/13013>

- 각각의 i 에 대해서, $S[SA[i]] < S[SA[i+1]]$ 가 되게 S 를 만들 수 있다.
- $SA = (2, 3, 1, 0)$ 인 경우에
- $SA[0] = 2$ 이기 때문에, $S[2] = a$ 를 넣고
- $SA[1] = 3$ 이기 때문에, $S[3] = b$ 를 넣고
- $SA[2] = 1$ 이기 때문에, $S[1] = c$ 를 넣고
- $SA[3] = 0$ 이기 때문에, $S[0] = d$ 를 넣는다.

접미사 배열 2

<https://www.acmicpc.net/problem/13013>

- $S[SA[i]] \leq S[SA[i+1]]$ 인 경우에 문제를 풀어본다
- $SA[0]$ 은 아무 제약 조건이 없기 때문에, 그냥 $S[SA[0]] = a$ 를 넣는다
- $SA[1]$ 은 두 가지 경우가 가능하다
- $S[SA[0]] < S[SA[1]]$ 또는 $S[SA[0]] == S[SA[1]]$

접미사 배열 2

<https://www.acmicpc.net/problem/13013>

- 두 문자열의 비교는 앞에서 부터 한 글자씩 비교를 한다

접미사 배열 2

<https://www.acmicpc.net/problem/13013>

- 두 접미사의 비교는 문자 하나와 접미사 비교로 바꿀 수 있다

*i*번 접미사:

S[i]	S[i+1]			...
------	--------	--	--	-----

*j*번 접미사:

S[j]	S[j+1]			...
------	--------	--	--	-----

접미사 배열 2

<https://www.acmicpc.net/problem/13013>

$[2, 0, 1, 2]$

20

- 두 접미사의 비교는 문자 하나와 접미사 비교로 바꿀 수 있다
- $S[i]$ 와 $S[j]$ 를 비교한다

i번 접미사:

$S[i]$	$S[i+1]$...
--------	----------	--	--	-----

j번 접미사:

$S[j]$	$S[j+1]$...
--------	----------	--	--	-----

접미사 배열 2

<https://www.acmicpc.net/problem/13013>

- 두 접미사의 비교는 문자 하나와 접미사 비교로 바꿀 수 있다
- $i+1$ 번 접미사와 $j+1$ 번 접미사

i 번 접미사:

$S[i]$	$S[i+1]$...
--------	----------	--	--	-----

j 번 접미사:

$S[j]$	$S[j+1]$...
--------	----------	--	--	-----

문자열 S, t

① $S < t$
 $S[0] < t[0]$

② $S[0] == t[0]$

&& $S[1:] < t[1:]$

접미사 배열 2

<https://www.acmicpc.net/problem/13013>

- 두 접미사의 비교는 문자 하나와 접미사 비교로 바꿀 수 있다
- $i+1$ 번 접미사와 $j+1$ 번 접미사

$i+1$ 번 접미사:

$S[i+1]$...
----------	--	--	-----

$j+1$ 번 접미사:

$S[j+1]$...
----------	--	--	-----

접미사 배열 2

<https://www.acmicpc.net/problem/13013>

- 각각의 i 에 대해서
- $S[SA[i]] < S[SA[i+1]]$ 이면 새로운 문자를 추가해야 하는 것 이기 때문에
- 되도록
- $S[SA[i]] = S[SA[i+1]]$ 을 많이 사용하는 것이 좋다.

접미사 배열 2

<https://www.acmicpc.net/problem/13013>

- 각각의 i 에 대해서
- $S[SA[i]] < S[SA[i+1]]$ 을 최소로 하는 것은
- $S[SA[i+1]] = S[SA[i]] + 1$ 을 최소로 하는 것과 같기 때문에
- 사전 순으로 앞서는 문자열을 만들 수 있게 된다

접미사 배열 2

<https://www.acmicpc.net/problem/13013>

- $S[SA[i]] == S[SA[i+1]]$ 인 경우
- i 번째 접미사와 $i+1$ 번째 접미사를 비교하는 것은
- i 번째 문자와 $i+1$ 번째 글자를 비교하고 같은 경우에는
- $i+1$ 번째 접미사와 $i+2$ 번째 접미사를 비교하는 것과 같다

접미사 배열 2

<https://www.acmicpc.net/problem/13013>

- $S[SA[i]] == S[SA[i+1]]$ 인 경우
- i 번째 접미사와 $i+1$ 번째 접미사를 비교하는 것은
- i 번째 문자와 $i+1$ 번째 글자를 비교하고 같은 경우에는
- $i+1$ 번째 접미사와 $i+2$ 번째 접미사를 비교하는 것과 같다
- 지금 $S[SA[i]] == S[SA[i+1]]$ 인 경우라는 것은 i 번째 문자와 $i+1$ 번째 글자가 같다는 것이기 때문에
- $i+1$ 번째 접미사와 $i+2$ 번째 접미사를 비교해야 한다

접미사 배열 2

<https://www.acmicpc.net/problem/13013>

- 그런데
- 우리는 이미 접미사 배열을 가지고 있다
- 즉, 접미사 배열에서 $i+1$ 이 $i+2$ 보다 앞에 있으면 된다

접미사 배열 2

<https://www.acmicpc.net/problem/13013>

28

$\Rightarrow 7\text{번 접미사} < 4\text{번 접미사}$
① 7 글자 == 4번 글자
② 8번 접미사 < 5번 접미사 true

인덱스:

0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---

접미사 배열:

7	4	8	6	1	5	2	9	3	0
---	---	---	---	---	---	---	---	---	---

문자열 S:

				a			a		
--	--	--	--	---	--	--	---	--	--

- 7번 접미사가 제일 앞서야 한다.
- S의 7번째 글자는 a라고 할 수 있다

접미사 배열 2

<https://www.acmicpc.net/problem/13013>

29

4번 접 < 8번 접미사
① 4번 글자 == 8번 글자
② 5번 접 < 9번 접미사

인덱스:

0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---

접미사 배열:

7	4	8	6	1	5	2	9	3	0
---	---	---	---	---	---	---	---	---	---

문자열 S:

				a			a	a	
--	--	--	--	---	--	--	---	---	--

- 7번 접미사 < 4번 접미사
- 4번째 글자를 a라고 하면 8번 접미사와 5번 접미사를 비교해야 한다
- 8번 접미사의 위치: 2, 5번 접미사의 위치: 5
- 4번째 글자가 a여도 7번 접미사 < 4번 접미사이다

접미사 배열 2

<https://www.acmicpc.net/problem/13013>

30

8번 접 < 6번 접

① 8번글자 = 6번글자

② 9번 접 < 7번 접

인덱스:

0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---

접미사 배열:

7	4	8	6	1	5	2	9	3	0
---	---	---	---	---	---	---	---	---	---

문자열 S:

				a		b	a	a	
--	--	--	--	---	--	---	---	---	--

- 4번 접미사 < 8번 접미사
- 8번째 글자를 a라고 하면 5번 접미사와 9번 접미사를 비교해야 한다
- 5번 접미사의 위치: 5, 9번 접미사의 위치: 7
- 8번째 글자가 a여도 4번 접미사 < 8번 접미사이다

접미사 배열 2

<https://www.acmicpc.net/problem/13013>

인덱스:

0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---

접미사 배열:

7	4	8	6	1	5	2	9	3	0
---	---	---	---	---	---	---	---	---	---

문자열 S:

				a		b	a	a	
--	--	--	--	---	--	---	---	---	--

- 8번 접미사 < 6번 접미사
- 6번째 글자를 a라고 하면 9번 접미사와 7번 접미사를 비교해야 한다
- 9번 접미사의 위치: 7, 7번 접미사의 위치: 0
- 6번째 글자가 a이면 8번 접미사 < 6번 접미사가 될 수 없다. 따라서, 6번째 글자는 b이다

접미사 배열 2

<https://www.acmicpc.net/problem/13013>

인덱스:

0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---

접미사 배열:

7	4	8	6	1	5	2	9	3	0
---	---	---	---	---	---	---	---	---	---

문자열 S:

	b			a		b	a	a	
--	---	--	--	---	--	---	---	---	--

- 6번 접미사 < 1번 접미사
- 1번째 글자를 b라고 하면 7번 접미사와 2번 접미사를 비교해야 한다
- 7번 접미사의 위치: 0, 2번 접미사의 위치: 6
- 1번째 글자가 b여도 6번 접미사 < 1번 접미사이다

접미사 배열 2

<https://www.acmicpc.net/problem/13013>

인덱스:

0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---

접미사 배열:

7	4	8	6	1	5	2	9	3	0
---	---	---	---	---	---	---	---	---	---

문자열 S:

	b			a	c	b	a	a	
--	---	--	--	---	---	---	---	---	--

- 1번 접미사 < 5번 접미사
- 5번째 글자를 b라고 하면 2번 접미사와 6번 접미사를 비교해야 한다
- 2번 접미사의 위치: 6, 6번 접미사의 위치: 3
- 5번째 글자가 b이면 1번 접미사 < 5번 접미사가 될 수 없다. 따라서, 5번째 글자는 c이다

접미사 배열 2

<https://www.acmicpc.net/problem/13013>

인덱스:

0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---

접미사 배열:

7	4	8	6	1	5	2	9	3	0
---	---	---	---	---	---	---	---	---	---

문자열 S:

	b	c		a	c	b	a	a	
--	---	---	--	---	---	---	---	---	--

- 5번 접미사 < 2번 접미사
- 2번째 글자를 c라고 하면 6번 접미사와 3번 접미사를 비교해야 한다
- 6번 접미사의 위치: 3, 3번 접미사의 위치: 8
- 2번째 글자가 c여도 5번 접미사 < 2번 접미사이다

접미사 배열 2

<https://www.acmicpc.net/problem/13013>

인덱스:

0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---

접미사 배열:

7	4	8	6	1	5	2	9	3	0
---	---	---	---	---	---	---	---	---	---

문자열 S:

	b	c		a	c	b	a	a	d
--	---	---	--	---	---	---	---	---	---

- 2번 접미사 < 9번 접미사
- 9번째 글자를 c라고 하면 3번 접미사와 10번 접미사를 비교해야 한다
- 3번 접미사의 위치: 8, 10번 접미사는 존재하지 않는다
- 9번째 글자가 c이면 2번 접미사 < 9번 접미사가 될 수 없다. 따라서, 9번째 글자는 d이다

접미사 배열 2

<https://www.acmicpc.net/problem/13013>

인덱스:

0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---

접미사 배열:

7	4	8	6	1	5	2	9	3	0
---	---	---	---	---	---	---	---	---	---

문자열 S:

	b	c	d	a	c	b	a	a	d
--	---	---	---	---	---	---	---	---	---

- 9번 접미사 < 3번 접미사
- 3번째 글자를 d라고 하면 10번 접미사와 4번 접미사를 비교해야 한다
- 10번 접미사는 존재하지 않고, 4번 접미사의 위치: 1
- 3번째 글자가 d여도 9번 접미사 < 3번 접미사이다

접미사 배열 2

<https://www.acmicpc.net/problem/13013>

인덱스:

0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---

접미사 배열:

7	4	8	6	1	5	2	9	3	0
---	---	---	---	---	---	---	---	---	---

문자열 S:

d	b	c	d	a	c	b	a	a	d
---	---	---	---	---	---	---	---	---	---

- 3번 접미사 < 0번 접미사
- 0번째 글자를 d라고 하면 4번 접미사와 1번 접미사를 비교해야 한다
- 4번 접미사의 위치: 1, 1번 접미사의 위치: 4
- 0번째 글자가 d여도 3번 접미사 < 0번 접미사이다

접미사 배열 2

38

<https://www.acmicpc.net/problem/13013>

- 소스: <http://boj.kr/45c92a2a42d741c382b4e4bb5fca0421>

접미사 배열 1

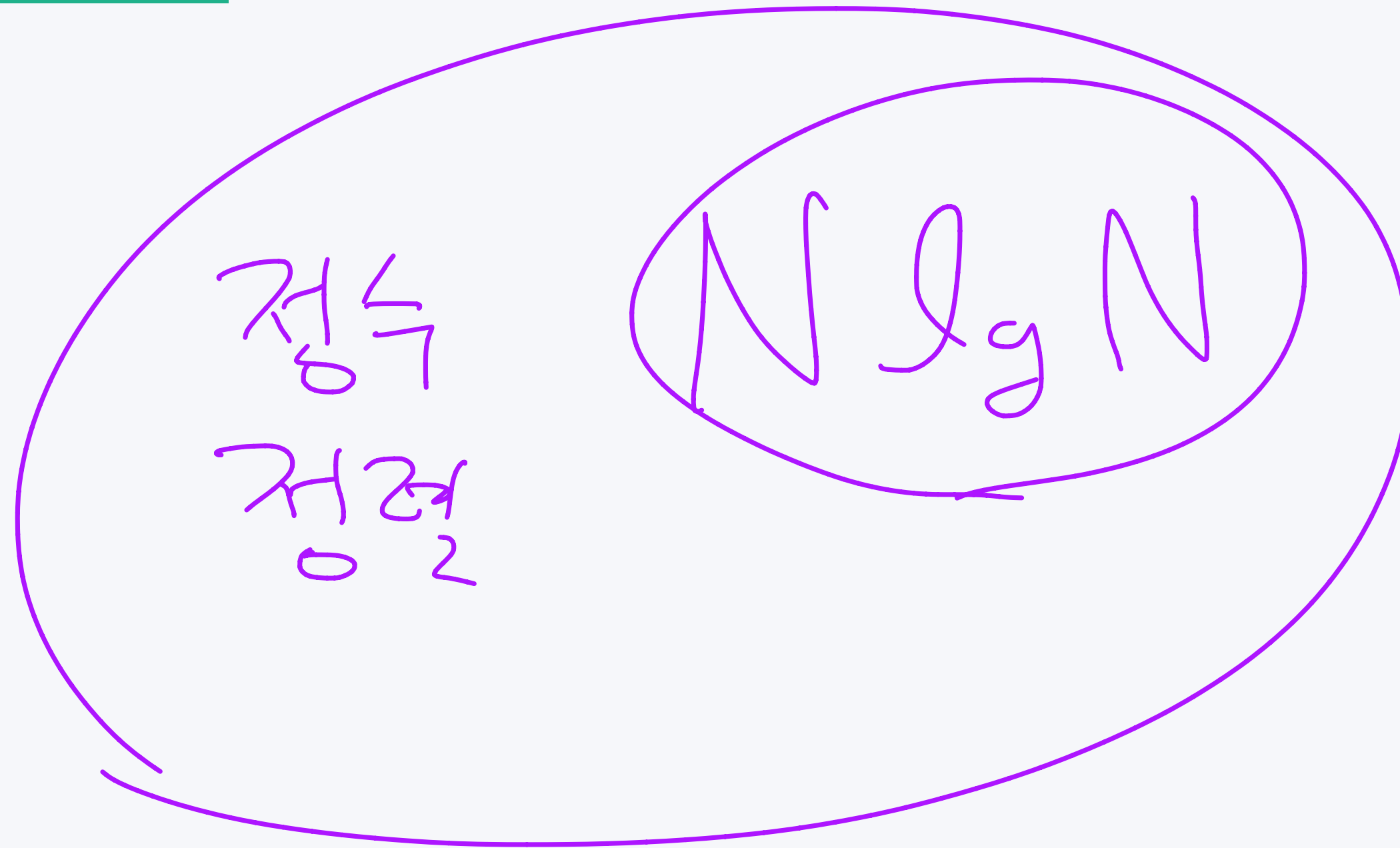
<https://www.acmicpc.net/problem/13012>

- 소스: <http://boj.kr/00cb0e6bc5e34cc5bce0adf8088cd43e>

접미사 배열

Suffix Array

- $O(N(\lg N)^2)$ 방법이 있다.
- 길이 1로 정렬
- 길이 2로 정렬
- 길이 4로 정렬
- ...



$\times \lg N$

$N \lg^2 N$

접미사 배열

Suffix Array

0: abcdabcabb

1: bcdabcabb

2: cdabcabb

3: dabcabb

4: abcabb

5: bcabb

6: cabb

7: abb

8: bb

9: b

인덱스:

0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---

문자열 S:

a	b	c	d	a	b	c	a	b	b
---	---	---	---	---	---	---	---	---	---

그룹 번호:

--	--	--	--	--	--	--	--	--	--

접미사 배열

Suffix Array

- 1글자를 기준으로 정렬하자

접미사 배열

Suffix Array

0	4: abccabb
	0: abcdabccabb
	7: abb
1	1: bcdabccabb
	8: bb
	5: bcabb
	9: b
2	2: cdabccabb
	6: cabb
3	3: dabccabb

인덱스:	0	1	2	3	4	5	6	7	8	9
문자열 S:	a	b	c	d	a	b	c	a	b	b
그룹 번호:	0	1	2	3	0	1	2	0	1	1

접미사 배열

Suffix Array

0	4: a
	0: a
	7: a
1	1: b
	8: b
	5: b
	9: b
2	2: c
	6: c
3	3: d

인덱스:

0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---

문자열 S:

a	b	c	d	a	b	c	a	b	b
---	---	---	---	---	---	---	---	---	---

그룹 번호:

0	1	2	3	0	1	2	0	1	1
---	---	---	---	---	---	---	---	---	---

- 1글자로 정렬이 모두 되어있는 상태이다
- 1글자로 정렬을 할 때는, 첫 번째 글자만 같으면 된다

접미사 배열

Suffix Array

0	4: abcabbb
	0: abcdabcabb
	7: abb
1	1: bcdabcabb
	8: bb
	5: bcabb
	9: b
2	2: cdabcabb
	6: cabb
3	3: dabcabb

인덱스:	0	1	2	3	4	5	6	7	8	9
문자열 S:	a	b	c	d	a	b	c	a	b	b
그룹 번호:	0	1	2	3	0	1	2	0	1	1

- 2글자를 기준으로 정렬을 해보자

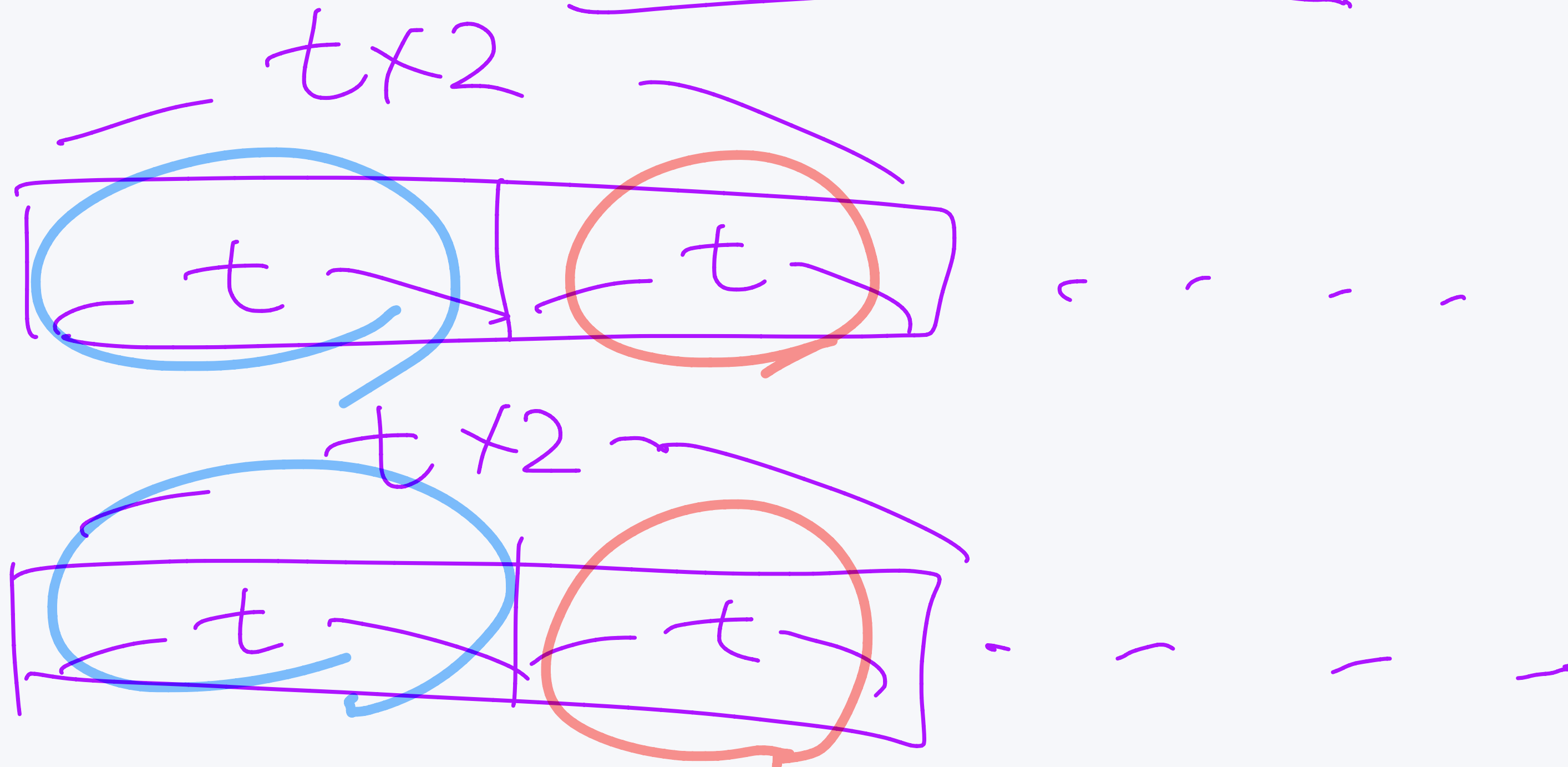
접미사 배열

Suffix Array

- 문자열 S 의 접미사의 접미사도 접미사이다

길이 t 정렬

길이 $t \times 2$



접미사 배열

Suffix Array

0	4: ab cab b
	0: ab cd ab cab b
	7: ab b
1	1: b cd ab cab b
	8: b b
	5: b cab b
	9: b
2	2: c d ab cab b
	6: c ab b
3	3: d ab cab b

인덱스:

0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---

문자열 S:

a	b	c	d	a	b	c	a	b	b
---	---	---	---	---	---	---	---	---	---

그룹 번호:

0	1	2	3	0	1	2	0	1	1
---	---	---	---	---	---	---	---	---	---

- 2글자를 기준으로 정렬을 해보자
- 4번 접미사와 0번 접미사는 두 글자를 기준으로 정렬하면 누가 앞에 와야 할까?

4: ab**cab**b

0: ab**cd**ab**cab**b

접미사 배열

Suffix Array

0	4: abccabb
	0: abcdabccabb
	7: abb
1	1: bcdabccabb
	8: bb
	5: bcabb
	9: b
2	2: cdabccabb
	6: cabb
3	3: dabccabb

인덱스:

0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---

문자열 S:

a	b	c	d	a	b	c	a	b	b
---	---	---	---	---	---	---	---	---	---

그룹 번호:

0	1	2	3	0	1	2	0	1	1
---	---	---	---	---	---	---	---	---	---

- 2글자를 기준으로 정렬을 해보자
- 4번 접미사와 0번 접미사는 두 글자를 기준으로 정렬하면 누가 앞에 와야 할까?

4: ab

0: ab

- 같아야 한다
- 어떻게 알 수 있을까?

접미사 배열

Suffix Array

0	4: abccabb
	0: abcdabccabb
	7: abb
1	1: bcdabccabb
	8: bb
	5: bcabb
	9: b
2	2: cdabccabb
	6: cabb
3	3: dabccabb

인덱스:	0	1	2	3	4	5	6	7	8	9
문자열 S:	a	b	c	d	a	b	c	a	b	b
그룹 번호:	0	1	2	3	0	1	2	0	1	1

49

- 2글자를 기준으로 정렬을 해보자
- 4번 접미사와 0번 접미사는 두 글자를 기준으로 정렬하면 누가 앞에 와야 할까?

4: ~~ab~~ccabb

0: ~~abcd~~abccabb

- 4번 접미사의 그룹 번호: 0
- 0번 접미사의 그룹 번호: 0

접미사 배열

Suffix Array

0	4:	abcabb
	0:	abcdabcabb
	7:	abb
1	1:	bcdabcabb
	8:	bb
	5:	bcabb
	9:	b
2	2:	cdabcabb
	6:	cabb
3	3:	dabcabb

인덱스:	0	1	2	3	4	5	6	7	8	9
문자열 S:	a	b	c	d	a	b	c	a	b	b
그룹 번호:	0	1	2	3	0	1	2	0	1	1

50

- 2글자를 기준으로 정렬을 해보자
- 4번 접미사와 0번 접미사는 두 글자를 기준으로 정렬하면 누가 앞에 와야 할까?

5: ~~bcabb~~

1: ~~bcdabcabb~~

- 5번 접미사의 그룹 번호: 1
- 1번 접미사의 그룹 번호: 1

접미사 배열

Suffix Array

0	4: abccabb
	0: abcdabccabb
	7: abb
1	1: bcdabccabb
	8: bb
	5: bcabb
	9: b
2	2: cdabccabb
	6: cabb
3	3: dabccabb

인덱스:

0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---

문자열 S:

a	b	c	d	a	b	c	a	b	b
---	---	---	---	---	---	---	---	---	---

그룹 번호:

0	1	2	3	0	1	2	0	1	1
---	---	---	---	---	---	---	---	---	---

51

- 2글자를 기준으로 정렬을 해보자
- 4번 접미사와 0번 접미사는 두 글자를 기준으로 정렬하면 누가 앞에 와야 할까?

4: abccabb

0: abcdabccabb

- 2글자를 기준으로 정렬했을 때
- 두 접미사의 그룹 번호는 같다

접미사 배열

Suffix Array

0	4: abcabbb
	0: abcdabcabb
	7: abb
1	1: bcdabcabb
	8: bb
	5: bcabb
	9: b
2	2: cdabcabb
	6: cabb
3	3: dabcabb

인덱스:

0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---

문자열 S:

a	b	c	d	a	b	c	a	b	b
---	---	---	---	---	---	---	---	---	---

그룹 번호:

0	1	2	3	0	1	2	0	1	1
---	---	---	---	---	---	---	---	---	---

52

- 2글자를 기준으로 정렬을 해보자
- 0번 접미사와 1번 접미사는 두 글자를 기준으로 정렬하면 누가 앞에 와야 할까?

0: abcdabcabb

1: bcdabcabb

- 0번 접미사의 그룹 번호: 0

- 1번 접미사의 그룹 번호: 1

접미사 배열

Suffix Array

0	4: abccabb
	0: abcdabccabb
	7: abb
1	1: bcdabccabb
	8: bb
	5: bcabb
	9: b
2	2: cdabccabb
	6: cabb
3	3: dabccabb

인덱스:

0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---

문자열 S:

a	b	c	d	a	b	c	a	b	b
---	---	---	---	---	---	---	---	---	---

그룹 번호:

0	1	2	3	0	1	2	0	1	1
---	---	---	---	---	---	---	---	---	---

53

- 2글자를 기준으로 정렬을 해보자
- 0번 접미사와 1번 접미사는 두 글자를 기준으로 정렬하면 누가 앞에 와야 할까?

0: abcdabccabb

1: bcdabccabb

- 2글자를 기준으로 정렬했을 때
- 0번 접미사가 앞서야 한다

접미사 배열

Suffix Array

0	4: abccabb
	0: abcdabccabb
	7: abb
1	1: bcdabccabb
	8: bb
	5: bcabb
	9: b
2	2: cdabccabb
	6: cab
3	3: dabccabb

인덱스:

0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---

문자열 S:

a	b	c	d	a	b	c	a	b	b
---	---	---	---	---	---	---	---	---	---

그룹 번호:

0	1	2	3	0	1	2	0	1	1
---	---	---	---	---	---	---	---	---	---

- 2글자를 기준으로 정렬을 해보자
- 2번 접미사와 6번 접미사는 두 글자를 기준으로 정렬하면 누가 앞에 와야 할까?

2: cdabccabb

6: cab

- 2번 접미사의 그룹 번호: 2

- 6번 접미사의 그룹 번호: 2

접미사 배열

Suffix Array

0	4: abccabb
	0: abcdabccabb
	7: abb
1	1: bcdabccabb
	8: bb
	5: bcabb
	9: b
2	2: cdabccabb
	6: cabb
3	3: dabccabb

인덱스:

0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---

문자열 S:

a	b	c	d	a	b	c	a	b	b
---	---	---	---	---	---	---	---	---	---

그룹 번호:

0	1	2	3	0	1	2	0	1	1
---	---	---	---	---	---	---	---	---	---

- 2글자를 기준으로 정렬을 해보자
- 2번 접미사와 6번 접미사는 두 글자를 기준으로 정렬하면 누가 앞에 와야 할까?

3: dabccabb

7: abb

- 3번 접미사의 그룹 번호: 3
- 7번 접미사의 그룹 번호: 0

접미사 배열

Suffix Array

0	4: abccabb
	0: abcdabccabb
	7: abb
1	1: bcdabccabb
	8: bb
	5: bcabb
	9: b
2	2: cdabccabb
	6: cabb
3	3: dabccabb

인덱스:

0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---

문자열 S:

a	b	c	d	a	b	c	a	b	b
---	---	---	---	---	---	---	---	---	---

그룹 번호:

0	1	2	3	0	1	2	0	1	1
---	---	---	---	---	---	---	---	---	---

- 2글자를 기준으로 정렬을 해보자
- 2번 접미사와 6번 접미사는 두 글자를 기준으로 정렬하면 누가 앞에 와야 할까?

2: cdabccabb

6: cabb

- 2글자를 기준으로 정렬했을 때
- 6번 접미사가 앞서야 한다

접미사 배열

Suffix Array

0	4: abccabb
	0: abcdabccabb
	7: abb
1	1: bcdabccabb
	8: bb
	5: bcabb
	9: b
2	2: cdabccabb
	6: cabb
3	3: dabccabb

인덱스:

0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---

문자열 S:

a	b	c	d	a	b	c	a	b	b
---	---	---	---	---	---	---	---	---	---

그룹 번호:

0	1	2	3	0	1	2	0	1	1
---	---	---	---	---	---	---	---	---	---

- 2글자를 기준으로 정렬을 해보자
- 8번 접미사와 9번 접미사는 두 글자를 기준으로 정렬하면 누가 앞에 와야 할까?

8: bb

9: b

- 8번 접미사의 그룹 번호: 1
- 9번 접미사의 그룹 번호: 1

접미사 배열

Suffix Array

0	4: abccabb
	0: abcdabccabb
	7: abb
1	1: bcdabccabb
	8: bb
	5: bcabb
	9: b
2	2: cdabccabb
	6: cabb
3	3: dabccabb

인덱스:

0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---

문자열 S:

a	b	c	d	a	b	c	a	b	b
---	---	---	---	---	---	---	---	---	---

그룹 번호:

0	1	2	3	0	1	2	0	1	1
---	---	---	---	---	---	---	---	---	---

58

- 2글자를 기준으로 정렬을 해보자
- 8번 접미사와 9번 접미사는 두 글자를 기준으로 정렬하면 누가 앞에 와야 할까?

9: b

10:

- 9번 접미사의 그룹 번호: 1
- 10번 접미사의 그룹 번호: ?

접미사 배열

Suffix Array

0	4: abccabb
	0: abcdabccabb
	7: abb
1	1: bcdabccabb
	8: bb
	5: bcabb
	9: b
2	2: cdabccabb
	6: cabb
3	3: dabccabb

인덱스:

0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---

문자열 S:

a	b	c	d	a	b	c	a	b	b
---	---	---	---	---	---	---	---	---	---

그룹 번호:

0	1	2	3	0	1	2	0	1	1
---	---	---	---	---	---	---	---	---	---

59

- 10번 접미사처럼 문자열의 길이를 넘어가는 경우가 존재할 수 있다
- 이 경우 길이가 n 일 때, n 번 접미사를 비교하는 일만 일어나기 때문에
- 즉, $n+1$ 번 이상 접미사를 비교하는 일이 없기 때문에
- $\text{group}[n] = -1$ 로 넣어준다

접미사 배열

Suffix Array

0	4: abccabb
	0: abcdabccabb
	7: abb
1	1: bcdabccabb
	8: bb
	5: bcabb
	9: b
2	2: cdabccabb
	6: cabb
3	3: dabccabb

인덱스:	0	1	2	3	4	5	6	7	8	9
문자열 S:	a	b	c	d	a	b	c	a	b	b
그룹 번호:	0	1	2	3	0	1	2	0	1	1

- 2글자를 기준으로 정렬을 해보자
- 8번 접미사와 9번 접미사는 두 글자를 기준으로 정렬하면 누가 앞에 와야 할까?
- 9: b
- 10:
- 9번 접미사의 그룹 번호: 1
- 10번 접미사의 그룹 번호: -1

접미사 배열

Suffix Array

0	4: abccabb
	0: abcdabccabb
	7: abb
1	1: bcdabccabb
	8: bb
	5: bcabb
	9: b
2	2: cdabccabb
	6: cabb
3	3: dabccabb

인덱스:

0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---

문자열 S:

a	b	c	d	a	b	c	a	b	b
---	---	---	---	---	---	---	---	---	---

그룹 번호:

0	1	2	3	0	1	2	0	1	1
---	---	---	---	---	---	---	---	---	---

- 2글자를 기준으로 정렬을 해보자
- 8번 접미사와 9번 접미사는 두 글자를 기준으로 정렬하면 누가 앞에 와야 할까?

8: bb

9: b

- 2글자를 기준으로 정렬했을 때
- 9번 접미사가 앞서야 한다

접미사 배열

Suffix Array

0	4 :	abcabb
	0 :	abcdabcabb
	7 :	abb
1	9 :	b
2	8 :	bb
3	1 :	bcdabcabb
	5 :	bcabb
4	6 :	cabb
5	2 :	cdabcabb
6	3 :	dabcabb

인덱스:	0	1	2	3	4	5	6	7	8	9
문자열 S:	a	b	c	d	a	b	c	a	b	b
그룹 번호:	0	3	5	6	0	3	2	0	2	1

- 2글자를 기준으로 정렬을 해보자

접미사 배열

Suffix Array

0	4: ab
	0: ab
	7: ab
1	9: b
2	8: bb
3	1: bc
	5: bc
4	6: ca
5	2: cd
6	3: da

인덱스:

0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---

문자열 S:

a	b	c	d	a	b	c	a	b	b
---	---	---	---	---	---	---	---	---	---

그룹 번호:

0	3	5	6	0	3	2	0	2	1
---	---	---	---	---	---	---	---	---	---

- 2글자를 기준으로 정렬이 되어있는 상태이다

접미사 배열

Suffix Array

0	4: abccabb
	0: abcdabccabb
	7: abb
1	9: b
2	8: bb
3	1: bcdabccabb
	5: bcabb
4	6: cabb
5	2: cdabccabb
6	3: dabccabb

인덱스:

0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---

문자열 S:

a	b	c	d	a	b	c	a	b	b
---	---	---	---	---	---	---	---	---	---

그룹 번호:

0	3	5	6	0	3	2	0	2	1
---	---	---	---	---	---	---	---	---	---

- 4글자를 기준으로 정렬을 해보자

접미사 배열

Suffix Array

0	4: abccabb
	0: abcdabccabb
	7: abb
1	9: b
2	8: bb
3	1: bcdabccabb
	5: bcabb
4	6: cabb
5	2: cdabccabb
6	3: dabccabb

인덱스:

0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---

문자열 S:

a	b	c	d	a	b	c	a	b	b
---	---	---	---	---	---	---	---	---	---

그룹 번호:

0	3	5	6	0	3	2	0	2	1
---	---	---	---	---	---	---	---	---	---

- 4글자를 기준으로 정렬을 해보자
- 4번 접미사와 0번 접미사는 두 글자를 기준으로 정렬하면 누가 앞에 와야 할까?

4: abccabb

0: abcdabccabb

- 4번 접미사의 그룹 번호: 0
- 0번 접미사의 그룹 번호: 0

접미사 배열

Suffix Array

0	4: abccabb
	0: abcdabccabb
	7: abb
1	9: b
2	8: bb
3	1: bcdabccabb
	5: bcabb
4	6: cabb
5	2: cdabccabb
6	3: dabccabb

인덱스:

0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---

문자열 S:

a	b	c	d	a	b	c	a	b	b
---	---	---	---	---	---	---	---	---	---

그룹 번호:

0	3	5	6	0	3	2	0	2	1
---	---	---	---	---	---	---	---	---	---

- 4글자를 기준으로 정렬을 해보자
- 4번 접미사와 0번 접미사는 두 글자를 기준으로 정렬하면 누가 앞에 와야 할까?

6: cabb

2: cdabccabb

- 6번 접미사의 그룹 번호: 2
- 2번 접미사의 그룹 번호: 5

접미사 배열

Suffix Array

0	4: abccabb 0: abcdabccabb 7: abb
1	9: b
2	8: bb
3	1: bcdabccabb 5: bcabb
4	6: cabb
5	2: cdabccabb
6	3: dabccabb

인덱스:

0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---

문자열 S:

a	b	c	d	a	b	c	a	b	b
---	---	---	---	---	---	---	---	---	---

그룹 번호:

0	3	5	6	0	3	2	0	2	1
---	---	---	---	---	---	---	---	---	---

- 4글자를 기준으로 정렬을 해보자
- 4번 접미사와 0번 접미사는 두 글자를 기준으로 정렬하면 누가 앞에 와야 할까?

4: abccabb

0: abcdabccabb

- 2글자를 기준으로 정렬했을 때
- 4번 접미사가 앞서야 한다

접미사 배열

Suffix Array

0 | 7: abb
1 | 4: abcabb
2 | 0: abcdabcabb
3 | 9: b
4 | 8: bb
5 | 5: bcabb
6 | 1: bcdabcabb
7 | 6: cabb
8 | 2: cdabcabb
9 | 3: dabcabb

인덱스:

0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---

문자열 S:

a	b	c	d	a	b	c	a	b	b
---	---	---	---	---	---	---	---	---	---

그룹 번호:

2	6	8	9	1	5	7	0	4	3
---	---	---	---	---	---	---	---	---	---

- 4글자를 기준으로 정렬을 해보자

접미사 배열

Suffix Array

0 | 7: abb
1 | 4: abca
2 | 0: abcd
3 | 9: b
4 | 8: bb
5 | 5: bcab
6 | 1: bcda
7 | 6: cabb
8 | 2: cdab
9 | 3: dabc

인덱스:

0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---

문자열 S:

a	b	c	d	a	b	c	a	b	b
---	---	---	---	---	---	---	---	---	---

그룹 번호:

2	6	8	9	1	5	7	0	4	3
---	---	---	---	---	---	---	---	---	---

- 4글자를 기준으로 정렬이 되어있는 상태

접미사 배열

Suffix Array

- t글자를 기준으로 정렬되어 있다면, t*2글자를 기준으로도 정렬할 수 있다
- N개의 문자열을 정렬하는데 걸리는 시간: $O(N \lg N)$
- 그런데, 문자열 비교가 아니고 정수 비교 2번으로 정렬할 수 있다.
- 따라서, $O(N^2 \lg N)$ 이 아니고 $O(N \lg N)$
- 정렬은 총 $\lg N$ 번 반복된다.
- 따라서, $O(N(\lg N)^2)$ 이다.

$$\frac{N \lg N \times \lg N}{\lg 2} = \frac{N \lg^2 N}{\lg 2}$$

접미사 배열 2

<https://www.acmicpc.net/problem/13264>

- 소스 1: <http://boj.kr/e229a80a10bc49b091c9c816c13824a6>
- 소스 2: <http://boj.kr/105c689a19904beb8dc2162a23cd442e>

서로 다른 부분 문자열의 개수

<https://www.acmicpc.net/problem/11478>

- 모든 부분 문자열은 suffix의 prefix 이다.
- abcde의 서로 다른 부분 문자열은 몇 개일까?

서로 다른 부분 문자열의 개수

<https://www.acmicpc.net/problem/11478>

- 모든 부분 문자열은 suffix의 prefix 이다.
- abcd의 서로 다른 부분 문자열은 몇 개일까?
- 10개
- abcd
- bcd
- cd
- d

서로 다른 부분 문자열의 개수

<https://www.acmicpc.net/problem/11478>

- 모든 부분 문자열은 suffix의 prefix 이다.
- abcd의 서로 다른 부분 문자열은 몇 개일까?
- 10개
- abcd (a, ab, abc, abcd)
- bcd (b, bc, bcd)
- cd (c, cd)
- d (d)

서로 다른 부분 문자열의 개수

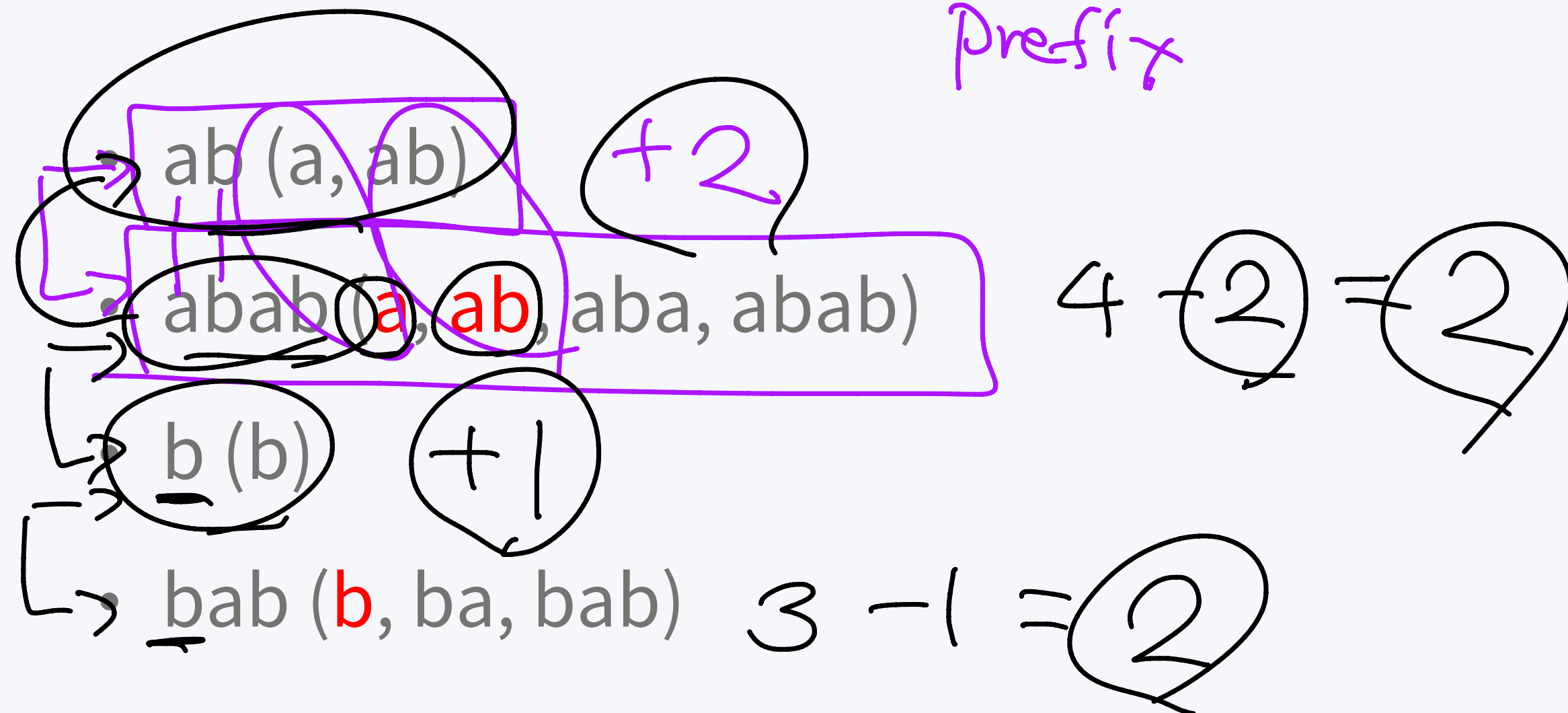
<https://www.acmicpc.net/problem/11478>

- 모든 부분 문자열은 suffix의 prefix 이다.

- abab의 서로 다른 부분 문자열은 몇 개일까?

- 7개

LCP: Longest
Common
Prefix



서로 다른 부분 문자열의 개수

<https://www.acmicpc.net/problem/11478>

- prefix가 같은 것의 개수를 빼줘야 한다.
- suffix array는 사전순 정렬이기 때문에, 바로 인접한 것과 비교를 해야 한다.
- i 와 $i-1$ 이 prefix가 몇개 까지 겹치는 지를 세어줘야 한다.

서로 다른 부분 문자열의 개수

77

<https://www.acmicpc.net/problem/11478>

- 소스: <http://boj.kr/781281af2b0f4a13be17f0447e573aa3>

LCP

Longest Common Prefix

- 가장 긴 prefix의 길이
- 앞에서 구한 P배열을 이용하면 LCP를 $\lg N$ 만에 구할 수 있다.
- 두 suffix i 와 $i+1$ 가 있을 때, 큰 k 부터 1씩 감소해나가면서 비교할 수 있다.

서로 다른 부분 문자열의 개수 2

<https://www.acmicpc.net/problem/11479>

- LCP를 $\lg N$ 만에 구하면 된다.

group [1] [5]
 5번 접미사를 길이 2를 기준으로
 정렬했을 때 순서

서로 다른 부분 문자열의 개수 2

80

<https://www.acmicpc.net/problem/11479>

- 소스: <http://boj.kr/97453cd4a562492590d36ce208b9dccb>

기수 정렬

Radix Sort

81

- 낮은 자리수 부터 비교하면서 정렬해나가는 방법

$$\frac{O(N \log N)}{\downarrow}$$
$$O(N)$$

기수 정렬

Radix Sort

82

- 260, 35, 25, 160, 8, 90, 33, 180, 23, 45, 10, 98을 정렬해보자

기수 정렬

Radix Sort

- 260, 35, 25, 160, 8, 90, 33, 180, 23, 45, 10, 98을 정렬해보자
 - 0: 260, 160, 90, 180, 10
 - 1:
 - 2:
 - 3: 33, 23
 - 4:
 - 5: 35, 25, 45
 - 6:
 - 7:
 - 8: 8, 98
 - 9:
-

기수 정렬

Radix Sort

84

- 0: 260, 160, 90, 180, 10
- 1:
- 2:
- 3: 33, 23
- 4:
- 5: 35, 25, 45
- 6:
- 7:
- 8: 8, 98
- 9:
- 결과: 260, 160, 90, 180, 10, 33, 23, 35, 25, 45, 8, 98

기수 정렬

Radix Sort

85

- 260, 160, 90, 180, 10, 33, 23, 35, 25, 45, 8, 98을 정렬해보자
- 0: 8
- 1: 10
- 2: 23, 25
- 3: 33
- 4: 45
- 5:
- 6: 260, 160
- 7:
- 8: 180
- 9: 90, 98

기수 정렬

Radix Sort

86

- 0: 8
- 1: 10
- 2: 23, 25
- 3: 33
- 4: 45
- 5:
- 6: 260, 160
- 7:
- 8: 180
- 9: 90, 98
- 결과: 8, 10, 23, 25, 33, 45, 260, 160, 180, 90, 98

기수 정렬

Radix Sort

87

- 8, 10, 23, 25, 33, 45, 260, 160, 180, 90, 98을 정렬해보자
- 0: 8, 10, 23, 25, 33, 45, 90, 98
- 1: 160, 180
- 2: 260
- 3:
- 4:
- 5:
- 6:
- 7:
- 8:
- 9:

기수 정렬

Radix Sort

88

- 0: 8, 10, 23, 25, 33, 45, 90, 98
- 1: 160, 180
- 2: 260
- 3:
- 4:
- 5:
- 6:
- 7:
- 8:
- 9:
- 결과: 8, 10, 23, 25, 33, 45, 90, 98, 160, 180, 260

기수 정렬

Radix Sort

89

- 기수 정렬의 시간 복잡도는 $O(dN)$ 이다
- 여기서 N 은 수의 개수, d 는 수의 자리



접미사 배열

Suffix Array

90

- t 글자를 기준으로 정렬되어 있다면, $t*2$ 글자를 기준으로도 정렬할 수 있다
- N 개의 문자열을 기수 정렬하는데 걸리는 시간: $O(N)$
- 두 번 정렬해야 하기 때문에 $O(N)$
- 정렬은 총 $\lg N$ 번 반복된다.
- 따라서, $O(N \lg N)$ 이다.

접미사 배열 2

<https://www.acmicpc.net/problem/13264>

- 소스: <http://boj.kr/8fad8e04dd5e402bb8399c1c451be076>

접미사 배열

Suffix Array

92

- $O(N)$ 만에 접미사 배열을 구현할 수 있다
- https://github.com/kcm1700/algorithms/blob/master/string/suffix_array.cpp

Suffix Array

<https://www.acmicpc.net/problem/9248>

- 접미사 배열과 LCP 배열을 만드는 문제

Suffix Array

<https://www.acmicpc.net/problem/9248>

- $LCP[i] = i$ 번째 Suffix와 $i-1$ 번째 Suffix의 LCP 길이

Suffix Array

<https://www.acmicpc.net/problem/9248>

- 소스: <http://boj.kr/496d6e22d77749ae8458a878b43aebbe>

Cubeditor

<https://www.acmicpc.net/problem/1701>

LCF의 최댓값 96

- 소문자 5,000개 이하로 구성된 문자열 S가 주어진다
- S의 부분 문자열은 연속된 일부분
- 두 번 이상 등장하는 부분 문자열 중에서 가장 긴 것의 길이?

Cubeditor

<https://www.acmicpc.net/problem/1701>

- 모든 부분 문자열은
- Suffix의 Prefix이다

	부분 문자열	
--	--------	--

Cubeditor

98

<https://www.acmicpc.net/problem/1701>

- 즉, 모든 접미사를 구하고 LCP 길이의 최대값이 정답이 된다

Cubeditor

<https://www.acmicpc.net/problem/1701>

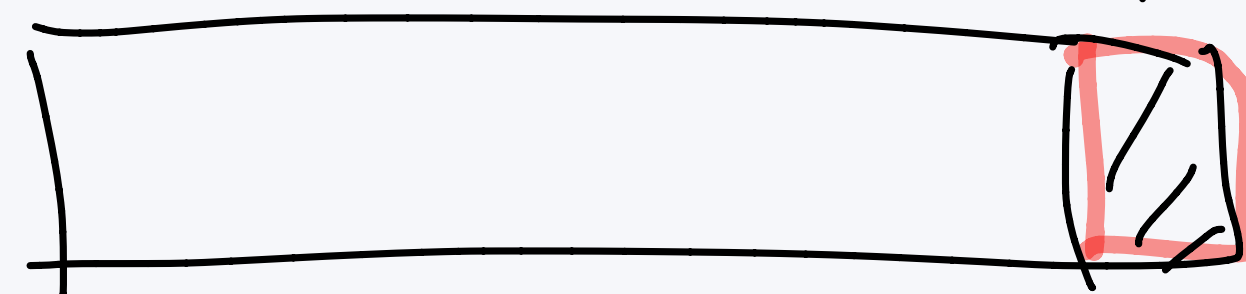
- 소스: <http://boj.kr/80be6aed2b494913b333e8f8bd82cebc>

LCS = Longest
Common

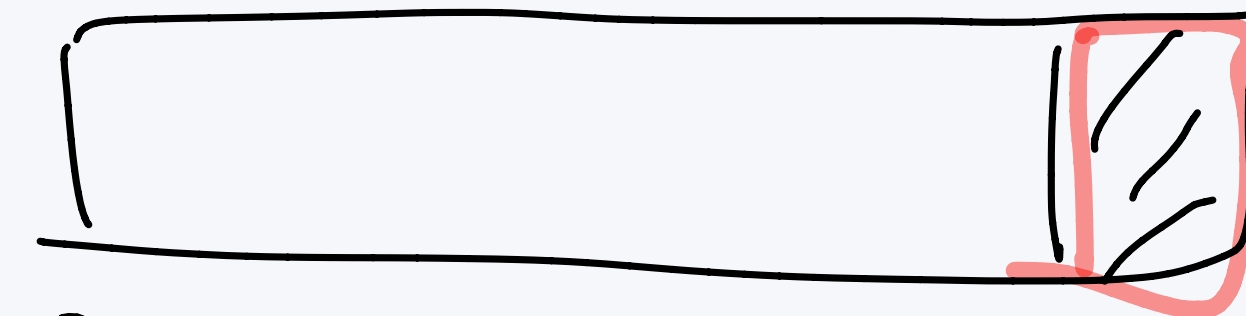
Sub Sequence

LCS \rightarrow Cubeditor
 \rightarrow Cubelover

A:



B:



$DP[N][M]$

$\text{if}(A[i] == B[j])$

$DP[i][j] = DP[i-1][j-1] + 1;$

$\text{else } DP[i][j] = \max(DP[i-1][j], DP[i][j-1])$

공통 부분 문자열

<https://www.acmicpc.net/problem/5582>

$DP[i][j] = A의 i까지$
 $B의 j까지$

100

- 두 문자열에 등장하는 부분 문자열 (연속) 중에서 가장 긴 것을 찾는 문제

공통 부분 문자열의
최대 길이

A: ABRACADABRA

B: ECADADABRBCRDARA

A: 

B: 

$\text{if } (A[i] == B[j])$

$DP[i][j] = DP[i-1][j-1] + 1$

$\text{else } DP[i][j] = 0$

공통 부분 문자열

101

<https://www.acmicpc.net/problem/5582>

- 두 문자열에 등장하는 부분 문자열 (연속) 중에서 가장 긴 것을 찾는 문제
- ABRACADABRA
- ECADADABRBCRDARA

공통 부분 문자열

102

<https://www.acmicpc.net/problem/5582>

- 두 문자열에 등장하는 부분 문자열 (연속) 중에서 가장 긴 것을 찾는 문제
- ABRACADABRA
- ECADADABRBCRDARA
- $D[i][j] = D[i-1][j-1] + 1 \text{ (} A[i] == A[j] \text{) else } 0$

공통 부분 문자열

103

<https://www.acmicpc.net/problem/5582>

- A
- ABRA
- ABRACADABRA
- ACADABRA
- ADABRA
- BRA
- BRACADABRA
- CADABRA
- DABRA
- RA
- RACADABRA
- ABRBC
- ADABRBC
- ADADABRBC
- BC
- BRBC
- C
- CADADABRBC
- DABRBC
- DADABRBC
- ECADADABRBC
- RBC

공통 부분 문자열

<https://www.acmicpc.net/problem/5582>

- A의 suffix x 에 대해서, s 보다 크면서 가장 작은 B의 suffix y 를 찾는다
- x 와 y 의 LCP가 공통 부분 문자열이 된다.
- 이 값 중에서 가장 큰 값을 찾으면 된다.

공통 부분 문자열

105

<https://www.acmicpc.net/problem/5582>

- 이 방법은 LCP를 구하는 것이 $O(N)$ 이다

공통 부분 문자열

abc, bcd

106

<https://www.acmicpc.net/problem/5582>

- 앞의 방법은 아래와 같이 두 문자열의 접미사 배열을 구하고 그것을 합친 것으로 볼 수 있다
- abc의 접미사 배열
 - abc
 - bc
 - c
- bcd의 접미사 배열
 - bcd
 - cd
 - d

공통 부분 문자열

<https://www.acmicpc.net/problem/5582>

- 앞의 방법은 아래와 같이 두 문자열의 접미사 배열을 구하고 그것을 합친 것으로 볼 수 있다

- abc의 접미사 배열

- abc

- bc

- c

- bcd의 접미사 배열

- bcd

- cd

- d

- abc와 bcd의 접미사 배열을 합친 결과

- abc

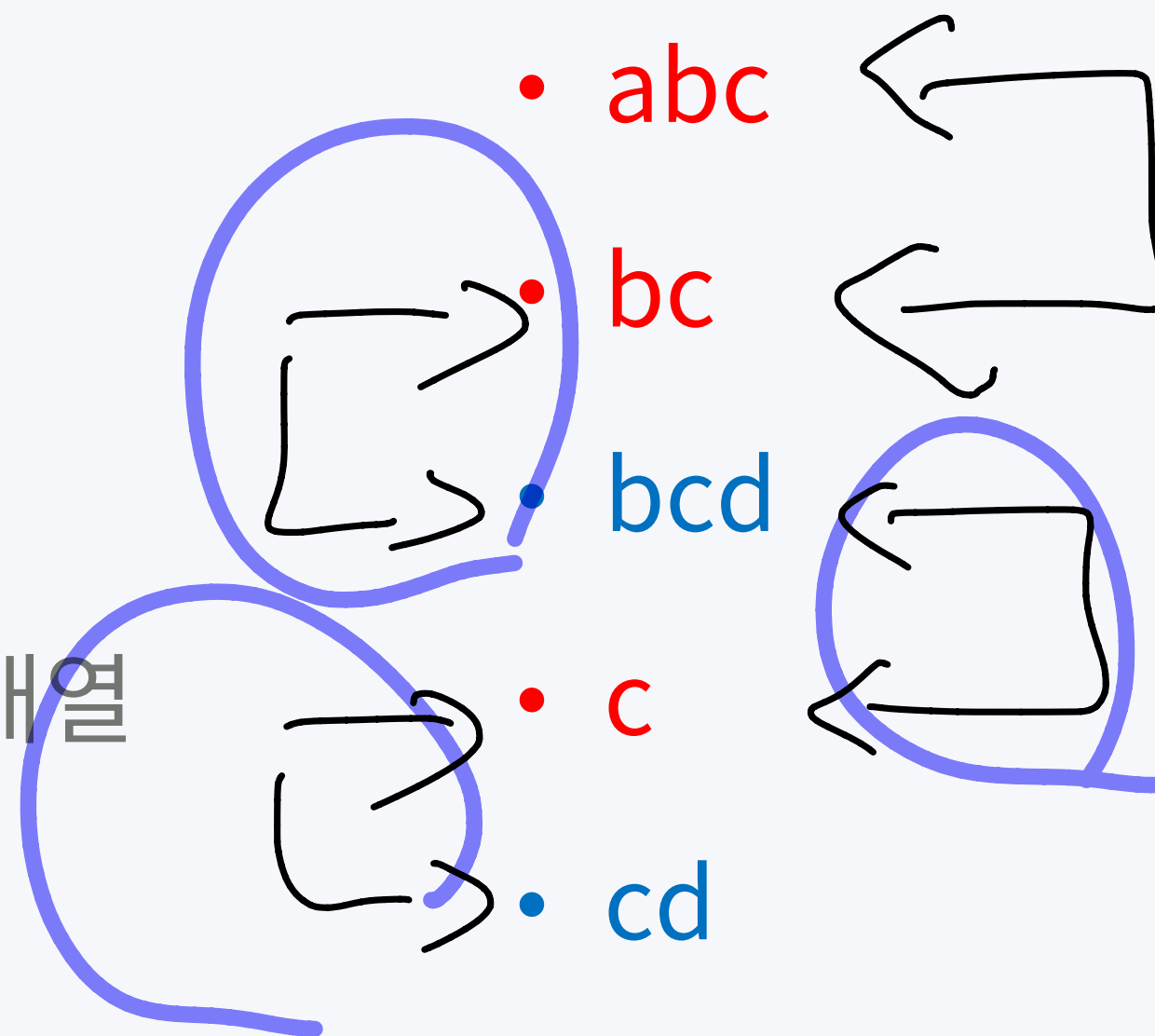
- bc

- bcd

- c

- cd

- d



공통 부분 문자열

<https://www.acmicpc.net/problem/5582>

108

abcd bcd
0 1 2 3 4 5

- 이것은 두 문자열을 합친 다음 접미사 배열을 구한 결과와 같다

- **abcbcd**의 접미사 배열
- abc와 bcd의 접미사 배열을 합친 결과

- **abcbcd** 0
- **bc** bcd 1
- **bcd** 3
- **cbcd**
- **cd**
- **d**

- **abc**
- **bc**
- **bcd**
- **c**
- **cd**
- **d**

2

공통 부분 문자열

109

<https://www.acmicpc.net/problem/5582>

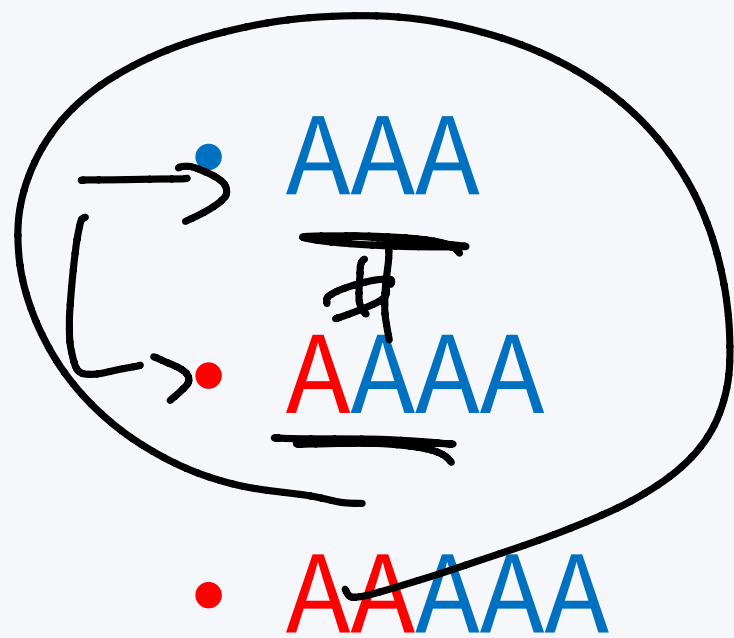
- 정답은 두 문자열을 합친 다음, 접미사 배열의 LCP의 최댓값이 된다

공통 부분 문자열

110

<https://www.acmicpc.net/problem/5582>

- 반례가 존재한다
- $A = AA$
- $B = AAA$
- $\overset{\#}{\checkmark} AA\overset{\checkmark}{A}AA$ 의 접미사
- A
- AA



공통 부분 문자열

111

<https://www.acmicpc.net/problem/5582>

- 사이에 A와 B에 등장할 수 없는 문자를 추가하면 된다
- $S = A + \text{'\#' } + B$

공통 부분 문자열

112

<https://www.acmicpc.net/problem/5582>

- 소스: <http://boj.kr/026a52f5197e4a799c3a8e3a9163795d>

Hidden Password

<https://www.acmicpc.net/problem/3789>

- 길이가 N인 문자열 S가 주어진다.
- 문자열 S의 가장 왼쪽 글자를 오른쪽에 붙여가면서 만들 수 있는 단어는 총 N개가 있다
- 예: S = alabala
- alabala
- labalaa
- abalaal
- balaala
- alaalab
- laalaba
- aalabal

Hidden Password

114

<https://www.acmicpc.net/problem/3789>

- 길이가 N 인 문자열 S 가 주어진다.
- 문자열 S 의 가장 왼쪽 글자를 오른쪽에 붙여가면서 만들 수 있는 단어는 총 N 개가 있다
- 이 때, 사전 순으로 가장 앞서는 단어를 찾는 문제

Hidden Password

115

<https://www.acmicpc.net/problem/3789>

- 글자의 길이를 2배로 한 다음, 접미사 배열을 만든다
- 접미사의 번호가 원래 문자열의 길이보다 작은 것 중 첫 번째가 정답이 된다.

Hidden Password

<https://www.acmicpc.net/problem/3789>

- $S = \text{dbac}$, $S*2 = \text{dbacdbac}$
- ac
- acdbac
- bac
- bacdbac
- c
- cdbac
- dbac
- dbacdbac

Hidden Password

117

<https://www.acmicpc.net/problem/3789>

- $S = \text{baba}$, $S*2 = \text{babababa}$
- a
- aba
- ababa
- abababa
- ba
- baba
- bababa
- babababa

Hidden Password

118

<https://www.acmicpc.net/problem/3789>

- 같을 때는, 인덱스가 뒤에 있는것을 먼저 오게 정렬하면 된다

Hidden Password

119

<https://www.acmicpc.net/problem/3789>

- 소스: <http://boj.kr/04e769e8c35a4db4975ec78feb7ccf8d>

가장 긴 팰린드롬 부분 문자열

120

<https://www.acmicpc.net/problem/13275>

- 문자열 S의 가장 긴 팰린드롬 부분 문자열을 찾는 문제

가장 긴 팰린드롬 부분 문자열

121

<https://www.acmicpc.net/problem/13275>

- 문자열 S 와 S' (S 를 뒤집은 것)을 만들고
- 공통 부분 문자열 문제와 비슷하게 풀면 된다

가장 긴 팰린드롬 부분 문자열

122

<https://www.acmicpc.net/problem/13275>

- 문자열 S 와 S' (S 를 뒤집은 것)을 만들고
- 공통 부분 문자열 문제와 비슷하게 풀면 된다

가장 긴 팰린드롬 부분 문자열

123

<https://www.acmicpc.net/problem/13275>

- $S = \text{banana}$
- $S' = \text{ananab}$
- $S + \text{'\#'} + S' = \text{banana\#ananab}$

가장 긴 팰린드롬 부분 문자열

<https://www.acmicpc.net/problem/13275>

- 6 #ananab
- 5 a#ananab (LCP = 0)
- 11 ab (LCP = 1)
- 3 ana#ananab (LCP = 1)
- 9 anab (LCP = 3)
- 1 anana#ananab (LCP = 3)
- 7 ananab (LCP = 5)
- 12 b (LCP = 0)
- 0 banana#ananab (LCP = 1)
- 4 na#ananab (LCP = 0)
- 10 nab (LCP = 2)
- 2 nana#ananab (LCP = 2)
- 8 nanab (LCP = 4)

가장 긴 팰린드롬 부분 문자열

125

<https://www.acmicpc.net/problem/13275>

- 소스: <http://boj.kr/9677c4349e3a46709d00eb0e2ca59d72>