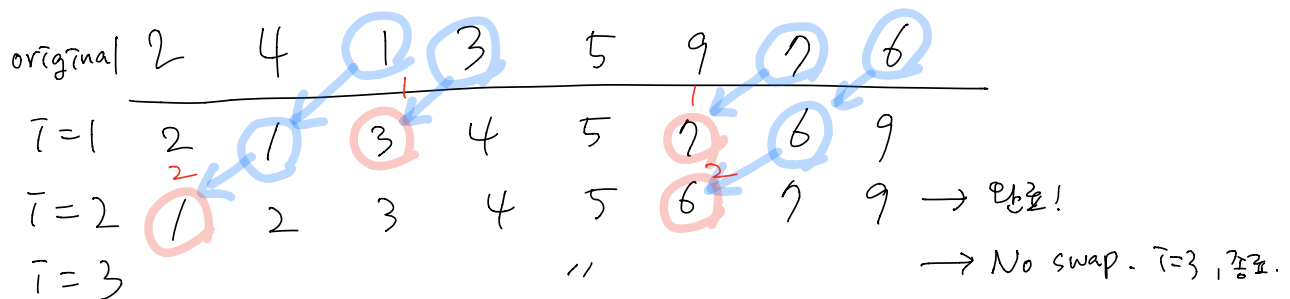


- 문제 : 배열이 주어질 때, 버블소트에서 바깥 for문에 대해 swap이 발생하는  $T$ 가 몇번인지  $\rightarrow$  swap이 더이상 일어나지 않을 때의  $T$ 를 출력하는 문제. ( $T$ 는 1부터 시작).

$$(1 \leq N \leq 500000), (0 \leq A[i] \leq 10^6)$$

- 접근 : 실제로 버블소트 실행하면  $N$ 범위 때문에 시간초과.



$\Rightarrow$  만약 9가 맨 앞에 있더라도 중요하지 않다.  
 $T$ 단계 한번 만에 맨 뒤로 가버리니까.

$\Rightarrow$  하지만, 작은 수가 앞으로 올 때는 한 단계마다 최대 1밖에 이동할 수 없다.

$\Rightarrow$  그럼 앞으로 땡겨온 횟수의 최대값을 구하면 되겠네!

but, 모든 경우를 체크할 수 있는가?

- 증명 : 가장 많이 앞칸으로 왔다는 말은 그보다 앞에 있는 수들중에 그보다 큰 수가 가장 많았다는 뜻.

(처음 ~) 각  $T$ 단계마다 연속해서 앞칸으로 이동했어야만 이걸 증명할 수 있다.  $\rightarrow$  ① 중간에 끊겼다가 다시 시작 ) 할 수 없다는 것을 보이면됨  
 ② 중간부터 시작

① ex) 4 7 9 1 3 이라는 배열이 있다고 할 때,

3을 기준으로 그보다 앞에 있는 더 큰 수는 3개가 있음.

$\Rightarrow$  버블소트는  $[i]$ 부터 시작해서 선형으로 '증가'하는 탐색

과정을 거치므로,

각 단계마다 3보다 앞에 있는 수들 중 1개는

무조건 3과 swap이 발생함.

② ex) 4 7 9 1 3 10 5 6 8

8은 기준으로 해도 처음부터 swap 발생.

⇒ 각 단계마다 가장 큰수가 맨 뒤로 이동하는 알고리즘.

그러므로 중간부터 시작한다는 것은

맨 처음 단계에서 그 수는 swap이 없었다는 뜻이고,

그보다 앞에 있던 최대값보다 크다는 뜻.

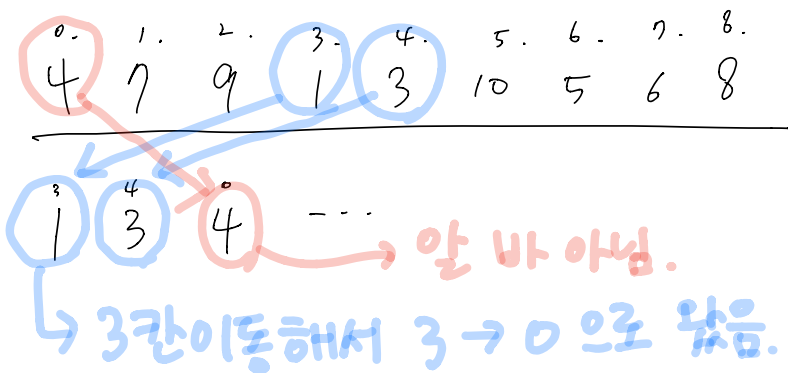
그것은 그 수가 최대값이고 그렇다면 다음 단계

에서도 앞의 더 큰수와 swap될 일은 없다는 뜻.

(모순.)

- 구현 : 원 배열 → 정렬된 배열에서 최대 '앞'으로 이동한 횟수를 어떻게 알 수 있을까?

원래의 인덱스와 정렬된 후의 인덱스를 비교하면 된다.



이러한 인덱스와 원소를 pair로 묶는다.

↓  
원소에 대해 정렬을 수행한다.

$$\Rightarrow \max_{i=0}^{n-1} (\text{원래 인덱스} - \text{현재 인덱스})$$

찾으면 +1한 값이 정답.

\* 문제에서  $1 \sim n+1$  까지 돌리는 이유는  
마지막  $n$ 번째에서 정렬이 완료됐을 경우를 위해  
한 번 더 수행.