

# 알고리즘 HW#1

2019-15861 염준영

April 11, 2021

## 1 Checker 프로그램의 작동

Checker 프로그램의 코드는 다음과 같다.

```
def check(arr, x, i):  
    low = high = 0  
    for j in range(0, len(arr)):  
        if (x < arr[j]):  
            high += 1  
        if (x > arr[j]):  
            low += 1  
    return low < i and len(arr) - high + 1 > i
```

$a_1, a_2, \dots, a_n$ 을 입력받았을 때, 그것을 정렬한 게  $b_1, b_2, \dots, b_n$ 이라고 하자. 우리가 찾은  $x$ 가 정렬된 순서에서  $i$ 번째( $1 \leq i \leq n$ )수가 맞는지를 확인하여야 한다. 즉  $x = b_i$ 인지를 확인해야 한다. 만약,  $b_{l+1} = \dots = b_m$ 으로 총  $m - l$ 개의 중복되는 숫자가 있다고 하면, ( $l < i \leq m$ , 즉 우리가 원하는  $b_i$ 의 값은 저 동일한 수들의 값과 같다.)  $b_n$ 을 다음과 같이 쓸 수 있다.

$$b_1, \dots, b_l, b_{l+1}, \dots, b_m, b_{m+1}, \dots, b_n$$

즉,  $b_i$ 보다 작은 수가  $l$ 개,  $b_i$ 보다 큰 수가  $n - m$ 개 이고,  $l < i < m + 1$ 이어야 한다. 따라서,  $x = b_i$ 이라면,  $x$ 보다 큰 것과,  $x$ 보다 작은 것의 개수를 세어, 각각을  $high$ ,  $low$ 라 하였을 때,  $low = l$ ,  $high = n - m$ 이므로,  $low < i < m + 1 =$

$(n - (n - m) + 1) = n - \text{high} + 1$ 인지 확인하면 되고, 이는 위의 코드와 같다. 이 코드는  $a_1, \dots, a_n$ 의 원소들을 한개씩  $x$ 와 비교하는데에  $O(n)$ 시간이 걸리고, 마지막에 boolean값을 계산하는데에  $O(1)$ 의 시간이 걸리므로 총  $O(n)$ 시간이 걸린다.

## 2 실행 시간의 비율

기본 스켈레톤 코드에 있는 1.in파일과, 테스트케이스 공유 게시판에 있는 (<http://etl.snu.ac.kr/mod/ubboard/article.php?id=1465974&bwid=2438616>, <http://etl.snu.ac.kr/mod/ubboard/article.php?id=1465974&bwid=2442082>) 테스트케이스들로 실행 시간을 측정하여 (randomized, deterministic)의 실행 시간  $(a_1, b_1), \dots, (a_{116}, b_{116})$ 을 얻고,  $b_i/a_i$ 의 평균을 구하여 2.096891053을 얻었다. 이 전체 평균의 경우, 대다수의(116개 중 100개)의 파일이 input의 크기가 100이었고, input파일의 크기가 특히 큰 경우(최소 1000개 100000개)의 경우에는  $b_i/a_i$ 의 값의 평균이 3.8이었다. 또한, input파일의 크기가 작을 때는 약 1.8부터 5.8까지, 들쭉날쭉한 경우가 많았고, 크기가 커질수록 비율이 들쭉날쭉하지 않고, 약 3.5전후의 값을 보였다. 이를 통해, deterministic linear time 알고리즘에서, 분할의 균형을 맞추기 위한 오버헤드가 크고, 일반적으로 randomized 알고리즘이 성능이 더 나음을 확인할 수 있었다. 또한, 사이즈 100개, 1000개, 10000개, 100000개의 각 경우, deterministic linear time 알고리즘은 실행 시간이 평균적으로 각각 약  $5.5 \times 10^{-5}$ ,  $5.5 \times 10^{-4}$ ,  $5.5 \times 10^{-3}$ ,  $5.5 \times 10^{-2}$ 초 정도로, 선형 시간을 따르고, 편차가 크지 않았고, randomized 알고리즘은 편차가 좀 더 컸다.

## 3 실행 환경

ETL에 있는 스켈레톤 코드를 받아, Python 3.8.5버전에서 실행하였다.