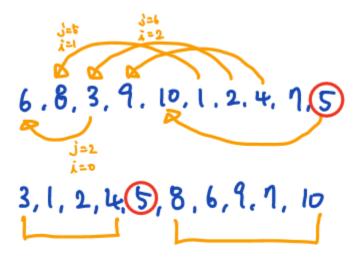
# 모두의 알고리즘 with 파이썬

문제 11 퀵 정렬

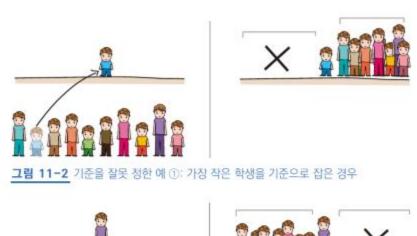
## 기준과 비교해서 그룹을 나눈 다음 각각 재귀 호출하여 합치는 방식

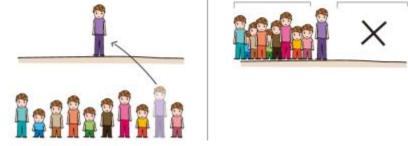


```
def quick_sort_sub(a, start, end):
   # 종료 조건: 정렬 대상이 한 개 이하이면 정렬할 필요가 없음
   if end - start <= 0:
      return
   # 기준 값을 정하고 기준 값에 맞춰 리스트 안에서 각 자료의 위치를 맞춤
   # [기준 값보다 작은 값들, 기준 값, 기준 값보다 큰 값들]
   pivot = a[end] # 편의상 리스트의 마지막 값을 기준 값으로 정함
  i = start
   for j in range(start, end):
      if a[j] <= pivot:</pre>
         a[i], a[j] = a[j], a[i]
         i += 1
   a[i], a[end] = a[end], a[i]
  # 재귀 호출 부분
  quick_sort_sub(a, start, i - 1) # 기준 값보다 작은 그룹을 재귀 호출로 다시 정렬
   quick_sort_sub(a, i + 1, end) # 기준 값보다 큰 그룹을 재귀 호출로 다시 정렬
def quick_sort(a):
   quick_sort_sub(a, 0, len(a) - 1)
d = [6, 8, 3, 9, 10, 1, 2, 4, 7, 5]
quick_sort(d)
print(d)
```



퀵정렬은 평균적으로 nlogn의 시간 복잡도를 가지지만, 최악의 경우 n^2의 시간 복잡도를 가진다. (최악의 경우는 기준이 최소값이나 최대값일 때)





**그림 11−3** 기준을 잘못 정한 예 ②: 가장 큰 학생을 기준으로 잡은 경우

#### • 평균

- ∘ 평균 T(n) = **O(nlog₂n)**
- 시간 복잡도가 O(nlog2n)를 가지는 다른 정렬 알고리즘과 비교했을 때도 가장 빠르다.
- 퀵 정렬이 불필요한 데이터의 이동을 줄이고 먼 거리의 데이터를 교환할 뿐만 아니라, 한 번 결정된 피벗들이 추후 연산에서 제외되는 특성 때문이다.

### 정렬 알고리즘 시간복잡도 비교

Name	Best	Avg	Worst	Run-time(정수60,000개) 단위: sec
삽입정렬	n	n <sup>2</sup>	n²	7.438
선택정렬	n²	n <sup>2</sup>	n²	10.842
버블정렬	n²	n <sup>2</sup>	n²	22.894
셸정렬	n	n <sup>1.5</sup>	n²	0.056
퀵정렬	nlog₂n	nlog₂ n	n²	0.014
힙 정렬	nlog₂n	nlog₂ n	nlog₂n	0.034
병합정렬	nlog₂n	nlog₂ n	nlog₂n	0.026

#### 연습 문제

- 11-1 지금까지 배운 네 가지 정렬 알고리즘 말고도 훨씬 많은 정렬 알고리즘이 있습니다. 그 중 하나인 거품 정렬(Bu bble sort)을 줄 서기로 비유하면 다음과 같습니다. 다음 과정을 읽고 리스트 [2, 4, 5, 1, 3]이 정렬되는 과정을 알고리즘으로 적어 보세요.
- 1 | 일단 학생들을 아무렇게나 일렬로 줄을 세웁니다.
- 2 | 선생님이 맨 앞에서부터 뒤로 이동하면서 이웃한 앞뒤 학생의 키를 서로 비교합니다. 앞에 있는 학생의 키가 바로 뒤에 있는 학생보다 크면 두 학생의 자리를 서로 바꿉니다.
- 3 | 선생님은 계속 뒤로 이동하면서 이웃한 앞뒤 학생의 키를 비교해서 필요하면 앞뒤 학생의 위치를 서로 바꿉니다.
- 4 | 모든 학생이 키 순서대로 줄을 설 때까지 이 과정을 반복합니다(줄의 끝까지 확인하는 동안 자리를 바꾼 적이 한 번도 없으면 모든 학생이 순서대로 줄을 선 것입니다).









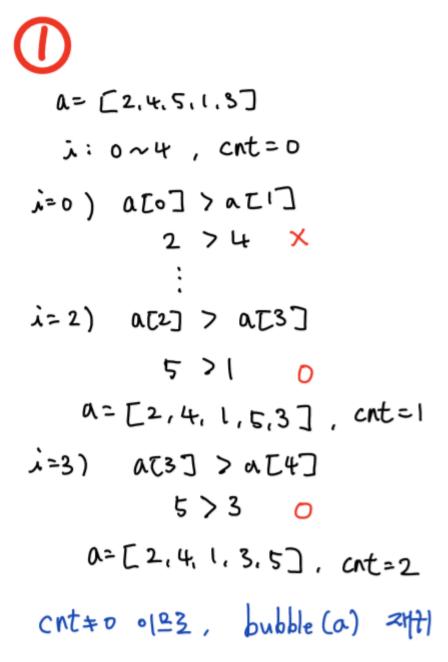
기준인 학생이 크다면 순서 바꾸고 기준 그대로

기준인 학생이 작다면 순서 그대로 기준 바꾸고

# 기준인 학생이 <del>크다면</del> 순서 바꾸고 기준 그대로 기준인 학생이 <del>작다면</del> 순서 그대로 기준 바꾸고

```
In [42]: def bubble(a):
             cnt=0
             for i in range(0,len(a)-1):
                                                    i = 선생님
                 if a[i]>a[i+1]:
                     a[i],a[i+1] = a[i+1],a[i]
                                                  a[*] = 학생
                     cnt +=1
             if cnt≕0:
                               ← 종료조건
                 return
             bubble(a)
In [44]: a = [2,4,5,1,3]
         bubble(a)
         print(a)
         [1, 2, 3, 4, 5]
```

```
def bubble(a):
    cnt=0
    for i in range(0,len(a)-1):
        if a[i]>a[i+1]:
            a[i],a[i+1] = a[i+1],a[i]
            cnt+=1
    if cnt=0:
        return
    bubble(a)
```



```
def bubble(a):
    cnt=0
    for i in range(0,len(a)-1):
        if a[i]>a[i+1]:
            a[i],a[i+1] = a[i+1],a[i]
            cnt+=1
    if cnt=0:
        return
    bubble(a)
```



$$A = [2,4,1,3,5]$$
 $A : 0 \sim 4$ ,  $Cnt = 0$ 
 $A = 0$ )  $A [0] > A [1]$ 
 $2 > 4 \times A$ 
 $A = 1$ 
 $A = 1$ 

```
def bubble(a):
    cnt=0
    for i in range(0,len(a)-1):
        if a[i]>a[i+1]:
            a[i],a[i+1] = a[i+1],a[i]
            cnt+=1
    if cnt=0:
        return
    bubble(a)
```



$$a = [2, 1, 3, 4, 5]$$
  
 $\downarrow$   
 $a = [1, 2, 3, 4, 5], cnt = 1$ 



$$\alpha = [1,2,3,4,5]$$
, cnt =0  
cnt = 0 0[23, return