

[Algorithm] 23강 : 퀵(quick) 정렬의 정의와 구현코드 — 나무늘보의 개발 블로그

노트북: 첫 번째 노트북

만든 날짜: 2020-11-09 오전 8:51

URL: <https://continuous-development.tistory.com/186>

Algorithm

[Algorithm] 23강 : 퀵(quick) 정렬의 정의와 구현 코드

2020. 11. 9. 08:49 수정 삭제 공개

퀵 정렬이란?

기존의 데이터를 설정하고 그 기준보다 큰 데이터와 작은 데이터의 위치를 바꾸는 방법

일반적인 상황에서 가장 많이 사용되는 정렬 알고리즘

병합 정렬과 더불어 대부분의 프로그래밍 언어의 정렬 라이브러리의 근간이 되는 알고리즘

가장 기본적인 퀵 정렬은 첫 번째 데이터를 기준 데이터(pivot)로 설정

퀵 정렬 동작 예시

1) step1

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

->

<-

1. 현재 피벗 값은 5이다.
2. 이때 왼쪽에서부터 5보다 큰 데이터를 선택하므로 7을 선택

3. 오른쪽에서부터 5보다 작은 데이터를 선택하므로 4를 선택
4. 이제 두 데이터의 위치를 서로 변경

2) step2

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

-> <-

이런 식으로 계속 진행하다 위치가 엇갈리는 경우가 발생한다.

왼쪽에서 5보다 큰 데이터를 선택하므로 6이 선택되고

오른쪽에서부터 5보다 작은 데이터를 선택하므로 1을 선택한다.

이처럼 위치가 엇갈리는 경우 피벗과 작은 데이터의 위치를 서로 변경한다.

3) step3

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

이제 5의 왼쪽에 있는 데이터는 5보다 작고 오른쪽에 있는 데이터는 모두 5보다 크다

이렇게 피벗을 기준으로 데이터 묶음으로 나누는 작업을 분할이라고 한다.

4) step4

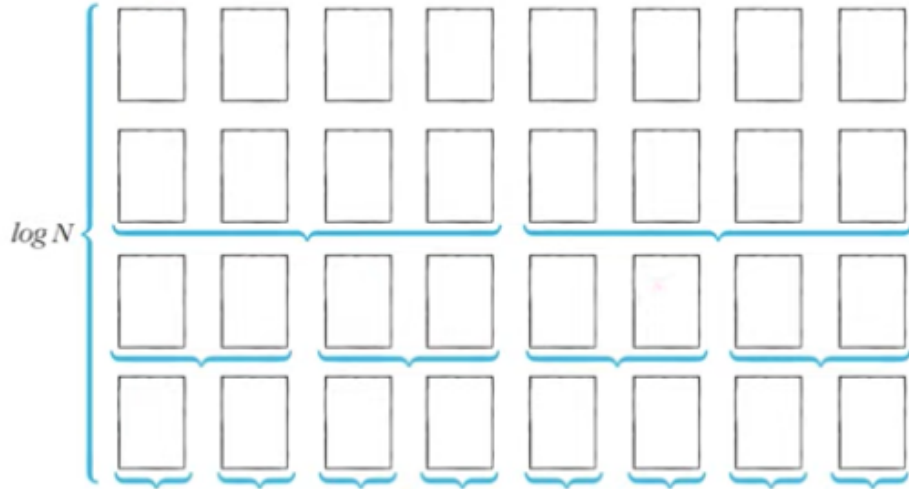
1		4		2		0		3
---	--	---	--	---	--	---	--	---

이제 분할된 데이터를 기준으로 피벗 값을 정하고 위에서 했던 것을 다시 반복한다.
이걸 재귀적으로 진행한다.

퀵 정렬이 빠른 이유

이상적인 경우 분할이 절반씩 이어 난다면 전체 연산 횟수로 $O(N \log N)$ 를 기대할 수 있다.

- 너비 X 높이 = $N \times \log N = N \log N$



퀵 정렬의 시간 복잡도

퀵 정렬은 평균의 경우 $O(N \log N)$ 의 시간 복잡도를 가진다.
하지만 최악의 경우 $O(N^2(\text{제곱}))$ 의 시간 복잡도를 가진다.

0 1 2 3 4 5 6 7 8 9

이와 같은 경우에 n 의 제곱의 시간 복잡도를 가진다.

퀵 정렬의 소스코드 - 일반적인 방식

```
array = [5,7,9,0,3,1,6,2,4,8]

def quick_sort(array,start,end):
    if start >= end: # 원소가 1개인 경우 종료
        return
    pivot = start # 피벗은 첫 번째 원소
    left = start + 1
    right = end
```

```

while (left <= right):
    # 피벗보다 큰 데이터를 찾을 때까지 반복
    while(left <= end and array[left] <= array[pivot]):
        left += 1
    # 피벗보다 작은 데이터를 찾을 때까지 반복
    while(left > end and array[left] >= array[pivot]):
        right += 1
    if (left > right): # 엇갈렸다면 작은 데이터와 피벗을 교체
        array[right], array[pivot] = array[pivot], array[right]
    else: # 엇갈리지 않았다면 작은 데이터와 큰 데이터를 교체
        array[left], array[pivot] = array[pivot], array[left]
    # 분할 이후 왼쪽 부분과 오른쪽 부분에서 각각 정렬 수행
    quick_sort(array, start, right-1)
    quick_sort(array, right+1, end )

quick_sort(array, 0 , len(array) -1)
print(array)
=> [0,1,2,3,4,5,6,7,8,9]

```

퀵 정렬의 소스코드 - python 방식

```

array = [5,7,9,0,3,1,6,2,4,8]

def quick_sort(array):
    if len(array) <= 1:
        return array
    pivot = array[0]
    tail = array[1:]

    left_side = [x for x in tail if x <= pivot] # 분할된 왼쪽 부분
    right_side = [x for x in tail if x > pivot] # 분할된 오른쪽 부분

    # 분할 이후 왼쪽 부분과 오른쪽 부분에서 각각 정렬 수행하고, 전체 리스트 반환
    return quick_sort(left_side) + [pivot] + quick_sort(right_side)

print(quick_sort(array))
=> [0,1,2,3,4,5,6,7,8,9]

```

www.youtube.com/watch?v=m-9pAwq1o3w&list=PLRx0vPvIEmdAghTr5mXQxGpHjWqSz0dgC

이 자료는 동빈 나 님의 이코 테 유튜브 영상을 보고 정리한 자료입니다.

'Algorithm' 카테고리의 다른 글

[Algorithm] 23강 : 퀵(quick) 정렬의 정의와 구현코드

[Algorithm] 22강 : 삽입 정렬의 정의와 구현코드

[Algorithm] 21강 : 선택 정렬의 정의와 구현코드

[Algorithm] 20강 : DFS & BFS 기초 문제 풀이

[Algorithm] 19강 : BFS(너비 우선 탐색) 알고리즘 정의와 예제

[Algorithm] 18강 : DFS(깊이 우선 탐색) 알고리즘 정의와 예제

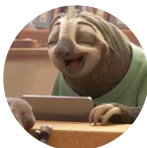
python 퀵정렬

quick sort

quick 정렬

퀵정렬

파이썬 퀵정렬



나무늘보스

혼자 끄적끄적하는 블로그 입니다.