

Algorithm/알고리즘

## [알고리즘](Python) 5가지 정렬 알고리즘: 선택, 버블, 병합, 퀵 정렬

그리버 | 2023. 10. 3. 01:30 | 수정 | 삭제

인천국제공항출발 뉴욕도착

982,584원 부터~

지금 보

서울출발 뉴욕도착

982,584원 부터~

지금 보

인천국제공항출발 밴쿠버도착

1,322,059원 부터~

지금 보

선택 정렬(Selection Sort)

선택 정렬의 개념

선택 정렬 코드 구현

선택 정렬의 장단점

버블 정렬(Bubble Sort)

병합 정렬(Merge Sort)

퀵 정렬의 개념

퀵 정렬(Quick Sort)

정렬의 개념

분체

풀이

정답 코드

정렬을 수행하는 방식에 따라 정렬 알고리즘이 달라진다.

그리고 그에 따라 연산량과 메모리 사용량 등의 성능이 달라지기 때문에

각 상황에 유리한 정렬 알고리즘을 알고 있어야 적재적소에 사용할 수 있다.

5가지 정렬 알고리즘을 비교해보자.

참고: <https://bang-tori.tistory.com/4>

### 1) 선택 정렬(Selection Sort)

#### 선택 정렬의 개념

무작위의 순서로 나열된 배열이 존재한다고 하자.

이 배열을 순회하면서 배열에서 가장 작은 요소를 찾아내면 맨 앞 요소와 교체한다.

정렬된 앞 부분을 제외한 나머지도 반복하여 교체한다.

이 과정이 선택 정렬이다.

패스	테이블	최솟값
0	[9,1,6,8,4,3,2,0]	0
1	[0,1,6,8,4,3,2,9]	1
2	[0,1,6,8,4,3,2,9]	2
3	[0,1,2,8,4,3,6,9]	3
4	[0,1,2,3,4,8,6,9]	4
5	[0,1,2,3,4,8,6,9]	6
6	[0,1,2,3,4,6,8,9]	8

#### 선택 정렬 코드 구현

Copy

```
1 def selectionSort(arr):
2     for i in range(len(arr)):
```

```

3     min_index = i
4
5     for j in range(i+1, len(arr)):
6         if arr[min_index] > arr[j]:
7             min_index = j
8
9     arr[i], arr[min_index] = arr[min_index], arr[i]
10
11 return arr

```

python

## 선택 정렬의 장단점

### 장점

- 알고리즘이 단순하다.
- 메모리를 적게 사용한다.

### 단점

- for 반복문을 두 번 사용하여 시간 복잡도가  $O(N^2)$ 로 높다.
  - 데이터의 개수  $n$ 이 커질 수록 성능이 매우 저하된다.

## 2) 버블 정렬(Bubble Sort)

### 버블 정렬의 개념

버블 정렬은 배열에서 두 수  $a, b$ 를 선택하고

그 두 수가 정렬되어 있다면 그대로 두고 정렬되어 있지 않다면 서로 바꾸는 방식이다.

이를 배열의 처음부터 끝까지 반복한다.

### 선택 정렬 코드 구현

```

1 def bubbleSort(arr):
2     for i in range(len(arr)-1):
3         for j in range(len(arr)-1-i):
4             if arr[j] > arr[j+1]:
5                 arr[j], arr[j+1] = arr[j+1], arr[j]
6
7     return arr

```

Copy

python

## 3) 병합 정렬(Merge Sort)

### 병합 정렬의 개념

병합 정렬은 분할 정복 알고리즘에 속한다.

### 분할 정복 알고리즘(Divide and conquer algorithm)

분할 정복 알고리즘은 해결할 수 없는 문제를 작은 문제로 분할해서 문제를 해결하는 방법이다.

퀵 정렬과 병합 정렬이 분할 정복 알고리즘에 속한다.

일반적으로 재귀 함수를 통해 분할 정복 알고리즘을 구현할 수 있다.

구조는 대개 아래와 같다.

```

1 function F(x):
2     if F(x)의 문제가 간단 then:
3         return F(x)를 직접 계산한 값
4     else:
5         x를 y1, y2로 분할
6         F(y1)과 F(y2)를 호출
7         return F(y1), F(y2)로부터 F(x)를 구한 값

```

Copy

python

그럼 병합 정렬 알고리즘을 살펴보자.

### n-way 병합 정렬

n-way 병합 정렬 알고리즘은 다음과 같다.

1. 정렬되지 않은 리스트를 각각 하나의 원소만 포함하는 n개의 부분리스트로 분할한다.
  - 한 원소만 들어있는 리스트는 이미 정렬된 것과 같으므로
2. 부분 리스트가 하나만 남을 때까지 반복해서 병합하며 정렬된 부분리스트를 생성한다.
3. 마지막으로 남은 부분리스트가 최종적으로 정렬을 마친 리스트이다.

## top-down 2-way 병합 정렬

일반적으로 n-way 병합 정렬의 개념을 사용하여 2-way 병합 정렬을 구현해 사용한다.

1. 리스트의 길이가 1 이하이면 이미 정렬된 것으로 본다.
2. 그렇지 않은 경우에는
  1. 분할(divide): 정렬되지 않은 리스트를 절반으로 잘라 비슷한 크기의 두 부분리스트로 나눈다.
  2. 정복(conquer): 각 부분 리스트를 재귀적으로 병합 정렬을 이용해 정렬한다.
  3. 결합(combine): 두 부분 리스트를 다시 하나의 정렬된 리스트로 병합한다. 이를 임시배열에 저장한다.
  4. 복사(copy): 임시배열에 저장된 결과를 원래 배열에 복사한다.

## 병합 정렬 코드 구현

```
1 def mergeSort(arr):
2     def sort(low, high):
3         if high - low < 2:
4             return
5         mid = (low + high) // 2
6         sort(low, mid)
7         sort(mid, high)
8         merge(low, mid, high)
9
10    def merge(low, mid, high):
11        temp = []
12        l, h = low, mid
13
14        while l < mid and h < high:
15            if arr[l] < arr[h]:
16                temp.append(arr[l])
17                l += 1
18            else:
19                temp.append(arr[h])
20                h += 1
21
22        while l < mid:
23            temp.append(arr[l])
24            l += 1
25        while h < high:
26            temp.append(arr[h])
27            h += 1
28
29        for i in range(low, high):
30            arr[i] = temp[i - low]
31
32    return sort(0, len(arr))
```

## 4) 퀵 정렬(Quick Sort)

참고: <https://www.daleseo.com/sort-quick/>

## 퀵 정렬의 개념

퀵 정렬은 분할 정복 알고리즘 방법으로 정렬을 수행한다.

## 퀵 정렬 코드 구현

```
1 def quick_sort(arr):
2     if len(arr) <= 1:
3         return arr
4     pivot = arr[len(arr) // 2]
5     lesser_arr, equal_arr, greater_arr = [], [], []
6     for num in arr:
7         if num < pivot:
8             lesser_arr.append(num)
9         elif num > pivot:
10            greater_arr.append(num)
11        else:
12            equal_arr.append(num)
13    return quick_sort(lesser_arr) + equal_arr + quick_sort(greater_arr)
```

## 문제

<https://www.acmicpc.net/problem/11650>



**11650번: 좌표 정렬하기**

첫째 줄에 점의 개수  $N$  ( $1 \leq N \leq 100,000$ )이 주어진다. 둘째 줄부터  $N$ 개의 줄에는  $i$ 번째 점의 위치  $x_i$ 와  $y_i$ 가 주어진다. ( $-100,000 \leq x_i, y_i \leq 100,000$ ) 좌표는 항상 정수이고, 위치가 같은 두 점은 없다.

www.acmicpc.net

주제	정렬
시간 / 메모리 제한	1 초 / 256 MB
정답 비율	47.915%

2차원 평면 위에 점  $N$ 개가 주어진다.

좌표를  $x$ 좌표가 증가하는 순으로,  $x$ 좌표가 같으면  $y$ 좌표가 증가하는 순으로 정렬한 다음 출력하는 프로그램을 작성하시오.

### 입력

- 첫째 줄에  $N$  ( $1 \leq N \leq 100,000$ )이 주어진다.
- 둘째 줄부터  $N$ 개의 줄에는  $i$ 번째 점의 위치  $x_i$ 와  $y_i$ 가 주어진다. ( $-100,000 \leq x_i, y_i \leq 100,000$ )
  - 좌표는 항상 정수이고, 위치가 같은 두 점은 없다.

### 출력

- 첫째 줄부터  $N$ 개의 줄에 점을 정렬한 결과를 출력한다.

- 예제 입력 1

```
1 5
2 3 4
3 1 1
4 1 -1
5 2 2
6 3 3
```

- 예제 출력 1

```
1 1 -1
2 1 1
3 2 2
4 3 3
5 3 4
```

## 풀이

우선 반복문을 이용해 입력되는  $x, y$  좌표를 2차원 배열에 저장한다.

파이썬에서는 `sort()` 또는 `sorted()` 함수를 통해 TimSort라는 삽입 정렬과 병합 정렬을 조합한 알고리즘으로 정렬을 수행할 수 있다.

`sorted()` 함수는 원본 리스트를 변형하지 않고 정렬된 결과를 반환 값으로 내보내는 반면,

`sort()` 함수는 원본 리스트를 변형한다.

어차피 정렬 후 원본 리스트를 다시 조회할 필요가 없기 때문에

`sort()` 함수를 사용하는 것이 메모리 측면에서 이득이기에

`sort()` 함수를 이용해 코드를 작성하였다.

2차원 배열에 대해 `sort()` 함수를 수행하면

0열에 따라 정렬을 하고 0열의 값이 같을 경우 1열에 대해 정렬을 수행하기 때문에

문제의 조건과 정확히 일치하게 된다.

## 정답 코드

```
1 import sys
2
3 N = int(sys.stdin.readline())
4 coor = []
5
6 for _ in range(N):
7     x, y = map(int, sys.stdin.readline().split())
8     coor.append([x, y])
9
10 coor.sort()
11 for c in coor:
12     print(c[0], c[1])
```

제출 번호	아이디	문제	결과	메모리	시간	언어	코드 길이	제출한 시간
67447977	aprilwldnjs	11650	맞았습니다!!	46764 KB	344 ms	Python 3 / 수정	195 B	24초 전



공감

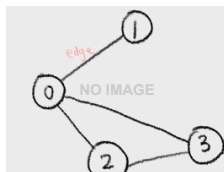
'Algorithm' > 알고리즘' 카테고리의 다른 글

- [BOJ](Python) 백준 2075번 N번째 큰 수 (0) 2023.09.26
- [알고리즘](Python) 그래프 개념과 파이썬 코드 구현 (3) 2023.09.19
- [알고리즘](Python) 백준 2696번 중앙값 구하기 - 우선순위 큐 (0) 2023.09.19
- [알고리즘](Python) 트리 (1) 2023.09.12
- [알고리즘](Python) 백준 10828번 - 스택 (0) 2023.09.12

'Algorithm/알고리즘' Related Articles



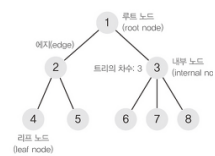
[BOJ](Python) 백준 2075  
번 N번째 큰 수



[알고리즘](Python) 그래프  
개념과 파이썬 코드 구현



[알고리즘](Python) 백준  
2696번 중앙값 구하기 - 우...



[알고리즘](Python) 트리

☐ Secret

안녕하세요! 어떤 댓글이든 환영합니다! 🍷

댓글달기

