

알고리즘 특강 정렬

자료구조에서 정렬을 공부하셨다면 알겠지만, 정말 중요한 내용입니다.
이번 시간엔 언어에서의 정렬을 주로 배우며, 활용은 다른 부분에서 볼 수 있습니다.

빠르게 복습하자.

Sorting

- FIFO (First In, First Out) 구조를 띄고 있는 자료구조로, 삽입과 삭제 연산이 서로 다른 한군데에서 발생함.
- 느린 알고리즘의 경우 시간복잡도가 $O(N^2)$, 빠른 경우 $O(\log N)$ 정도 된다.

C++에서의 정렬

```
#include <iostream>
#include <vector>
#include <algorithm>
using namespace std;
```

```
int main() {
    int arr[5] = {4, 6, 2, 1, 0};
    vector<int> vec = {9, 8, 7, 6, 5};
```

```
    sort(arr, arr + 5);
```

```
    sort(vec.begin(), vec.end());
```

```
    for(int i = 0; i < 5; i++) cout << arr[i] << endl;
```

```
    for(int i = 0; i < 5; i++) cout << vec[i] << endl;
```

```
    sort(arr, arr + 5, greater<int>());
```

```
    sort(vec.begin(), vec.end(), greater<int>());
```

```
    for(int i = 0; i < 5; i++) cout << arr[i] << endl;
```

```
    for(int i = 0; i < 5; i++) cout << vec[i] << endl;
```

```
}
```

- **template <class Rit>**
void sort(Rit first, Rit last)
시작 ~ 종료 전까지의 범위에 대하여 정렬함.

- **template <class Rit, class Compare>**
void sort(Rit first, Rit last, Compare comp)
정렬 기준을 활용해 정렬함.

greater<T>(): 내림차순 정렬
less<T>(): 오름차순 정렬 (Default)

C++에서의 정렬

```
#include <iostream>
#include <vector>
#include <algorithm>
using namespace std;
```

```
struct sample {
    int x, y;
};
```

```
bool compare(const sample &x, const sample &y) {
    if(x.x == y.x) return x.y < y.y;
    return x.x < y.x;
}
```

● 사용자 정의 비교 함수는 다음과 같이 선언함.
(출력이 0이면 값 교체)

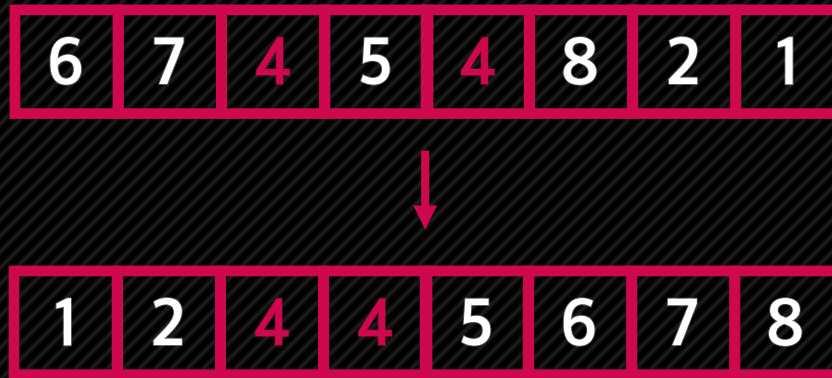
```
int main() {
    vector<sample> vec_sample = {{1, 2}, {3, 4}, {5, 6}};

    sort(vec_sample.begin(), vec_sample.end(), compare);

    for(auto s : vec_sample)
        cout << s.x << ' ' << s.y << endl;
}
```

● 마지막에 선언한 함수를 사용함.

Stable Sort



- 동일한 값의 데이터가 있을 때, **순서가 바뀌지 않음**이 보장되는가?

std::stable_sort



```
#include <iostream>
#include <vector>
#include <algorithm>
using namespace std;

int main() {
    int arr[5] = {4, 6, 2, 1, 0};
    vector<int> vec = {9, 8, 7, 6, 5};

    stable_sort(arr, arr + 5);
    stable_sort(vec.begin(), vec.end());

    for(int i = 0; i < 5; i++) cout << arr[i] << endl;
    for(int i = 0; i < 5; i++) cout << vec[i] << endl;

    stable_sort(arr, arr + 5, greater<int>());
    stable_sort(vec.begin(), vec.end(), greater<int>());

    for(int i = 0; i < 5; i++) cout << arr[i] << endl;
    for(int i = 0; i < 5; i++) cout << vec[i] << endl;
}
```

실습 (1)

Silver 5 - 수 정렬하기 (2)

요약

- N개의 수가 주어졌을 때, 이를 오름차순으로 정렬하는 프로그램을 작성하시오.

제약조건

- N의 범위는 $1 \leq N \leq 1,000,000$ 이다.
- 각각의 수의 절댓값의 범위는 $1 \leq N_i \leq 1,000,000$ 이다.
- 수는 중복되지 않는다.

실습 (2)

Silver 5 - 나이순 정렬

요약

- 온라인 저지에 가입한 사람들의 나이와 이름이 가입한 순서대로 주어진다.
- 나이가 증가하는 순으로, **나이가 같으면 먼저 가입한 사람이 앞에 오는 순서**로 정렬하는 프로그램을 작성하시오.

제약조건

- 총 인원의 범위는 $1 \leq N \leq 100,000$ 이다.

“Any question?”