

## Sort 메서스

### I 특징

- STL에서 제공하는 알고리즘
- Algorithm.h 에 위치
- Quick sort을 기반으로 함수가 구현되어있어, 평균 시간복잡도는  $n \log n$

### II 사용

- Sort(start, end)를 사용하여 [start, end)의 범위에 있는 인자(element)를 오름차순(default)으로 정렬해주는 함수 (start를 포함하고 end를 포함하지 않는 구간)

### III 원형

#### -기본

Template <typename T>

Void sort(T start, T end);

#### -정렬기준 사용

Template <typename T>

Void sort(T start, T end, Compare comp);

-3번째 인자를 넣지 않으면 default로 오름차순으로 정렬

-3번째 인자에 사용자가 정의한 함수를 기준으로 정렬을 할 수 있다(이항조건자를 이용 가능)

-Greater는 비교를 수행하기위한 함수 객체

#### IV 사용

##### -기본

```
1.  #include <iostream>
2.  #include <algorithm>
3.
4.  using namespace std;
5.
6.  void Print(int *arr)
7.  {
8.      cout << "arr[i] : " ;
9.      for(int i=0; i<10; i++)
10.     {
11.         cout << arr[i] << " ";
12.
13.     }cout << endl;}
14.
15.
16. int main() {
17.     int arr[10] = {3, 7, 2, 4, 1, 0, 9, 8, 5, 6};
18.     Print(arr);
19.     sort(arr, arr+10);
20.     Print(arr);
21.
22.     return 0;
23. }
```

##### 결과

```
arr[i] : 3 7 2 4 1 0 9 8 5 6
arr[i] : 0 1 2 3 4 5 6 7 8 9
```

-정렬기준 사용

```
1  #include <iostream>
2  #include <algorithm>
3
4  using namespace std;
5
6  void Print(int *arr)
7  {
8      cout << "arr[i] : " ;
9      for(int i=0; i<10; i++)
10     {
11         cout << arr[i] << " ";
12     }
13     cout << endl;
14
15
16  int main() {
17      int arr[10] = {3, 7, 2, 4, 1, 0, 9, 8, 5, 6};
18      Print(arr);
19      sort(arr, arr+10, greater<int>());
20      Print(arr);
21
22      return 0;
23  }
```

결과

arr[i] : 3 7 2 4 1 0 9 8 5 6

arr[i] : 9 8 7 6 5 4 3 2 1 0

-백터를 정렬

```
1  #include <iostream>
2  #include <algorithm>
3  #include <vector>
4  #include <ctime>
5
6  using namespace std;
7
8  void Print(vector<int> &v)
9  {
10     cout << "vector : " ;
11     for(int i=0; i<10; i++)
12     {
13         cout << v[i] << " ";
14     }
15     cout << endl;
16 }
17
18
19
20
21 int main() {
22     srand((int)time(NULL));
23
24     vector<int> v;
25     int n = 10;
26
27     for(int i=0; i<n; i++)
28     {
29         v.push_back(rand() % 10);
30     }
31
32     Print(v);
33     sort(v.begin(), v.end());
34     Print(v);
35     sort(v.begin(), v.end(), greater<int>());
36     Print(v);
37
38     return 0;
39 }
```

결과

vector : 1 7 5 9 4 5 6 0 1 0

vector : 0 0 1 1 4 5 5 6 7 9

vector : 9 7 6 5 5 4 1 1 0 0