

프로그래밍 언어 – STL (2)

최백준 choi@startlink.io

algorithm

count

STL

`count(begin, end, value)`

[begin, end)에 포함되어 있는 원소 중에서 `value`의 개수를 찾는다

`count_if(begin, end, p)`

[begin, end)에 포함되어 있는 원소 중에서 조건 `p`에 해당하는 것의 개수를 찾는다

시간복잡도: $O(N)$

count

STL

```
vector<int> a = {1, 4, 1, 2, 4, 2, 4, 2, 3, 4, 4};  
for (int i=1; i<=5; i++) {  
    cout << i << "의 개수: " << count(a.begin(), a.end(), i);  
    cout << '\n';  
}
```

1의 개수: 2

2의 개수: 3

3의 개수: 1

4의 개수: 5

5의 개수: 0

count

STL

```
vector<int> a = {1, 4, 1, 2, 4, 2, 4, 2, 3, 4, 4};  
int even = count_if(a.begin(), a.end(), [](int x) {  
    return x % 2 == 0;  
});  
int odd = count_if(a.begin(), a.end(), [](int x) {  
    return x % 2 == 1;  
});  
  
cout << "짝수의 개수: " << even << '\n';  
cout << "홀수의 개수: " << odd << '\n';
```

개수 세기

<https://www.acmicpc.net/problem/10807>

- <https://gist.github.com/Baekjoon/8a8cde8357d2b74414e0>

알파벳 개수

7

<https://www.acmicpc.net/problem/10808>

- <https://gist.github.com/Baekjoon/938052f312534091f3f4>

find

STL

`find(begin, end, value)`

[begin, end)에 포함되어 있는 원소 중에서 `value`의 이터레이터

`find_if(begin, end, p)`

[begin, end)에 포함되어 있는 원소 중에서 조건 `p`에 해당하는 것의 이터레이터

두 함수 모두 못 찾으면 `end`를 리턴

시간복잡도: $O(N)$

find

STL

```
vector<int> a = {1, 4, 1, 2, 4, 2, 4, 2, 3, 4, 4};  
for (int i=1; i<=5; i++) {  
    auto it = find(a.begin(), a.end(), i);  
    cout << i << "의 위치: ";  
    if (it == a.end()) {  
        cout << "찾을 수 없음";  
    } else {  
        cout << (it-a.begin());  
    }  
    cout << '\n';  
}
```

find

STL

1의 위치: 0

2의 위치: 3

3의 위치: 8

4의 위치: 1

5의 위치: 찾을 수 없음

find

STL

```
vector<int> a = {1, 4, 1, 2, 4, 2, 4, 2, 3, 4, 4};  
auto even = find_if(a.begin(), a.end(), [](int x) {  
    return x % 2 == 0;  
});  
auto odd = find_if(a.begin(), a.end(), [](int x) {  
    return x % 2 == 1;  
});  
  
cout << "첫 번째 짝수: " << (even - a.begin()) << '\n';  
cout << "첫 번째 홀수: " << (odd - a.begin()) << '\n';
```

알파벳 찾기

<https://www.acmicpc.net/problem/10809>

- <https://gist.github.com/Baekjoon/1bd4360bf97be15fb5a5>

fill

STL

`fill(begin, end, value)`
[begin, end)을 value로 채운다

시간복잡도: $O(N)$

fill

14

STL

```
vector<int> a = {1, 4, 1, 2, 4, 2, 4, 2, 3, 4, 4};  
for (int x : a) {  
    cout << x << ' '  
}  
cout << '\n';  
fill(a.begin(), a.end(), 0);  
for (int x : a) {  
    cout << x << ' '  
}  
cout << '\n';
```

공 냥기

<https://www.acmicpc.net/problem/10810>

- <https://gist.github.com/Baekjoon/b1ac23fec956727f3a9b>

reverse

STL

`reverse(begin, end)`

[begin, end)의 순서를 역순으로 만든다

시간복잡도: $O(N)$

reverse

STL

```
vector<int> a = {1, 4, 1, 2, 4, 2, 4, 2, 3, 4, 4};  
for (int x : a) {  
    cout << x << ' ';  
}  
cout << '\n';  
reverse(a.begin(), a.end());  
for (int x : a) {  
    cout << x << ' ';  
}  
cout << '\n';
```

바구니 뒤집기

18

<https://www.acmicpc.net/problem/10811>

- <https://gist.github.com/Baekjoon/32f1755a421ce0be5e7f>

rotate

STL

```
rotate(begin, mid, end)
```

[begin, end)을 mid를 기준으로 왼쪽으로 회전시킨다

begin에는 mid에 있던 값이

end-1에는 mid-1에 들어있던 값이 들어간다

시간복잡도: $O(N)$

rotate

STL

```
vector<int> a = {0, 1, 2, 3, 4, 5};  
rotate(a.begin(), a.begin()+2, a.end());  
for (int x : a) {  
    cout << x << ' ';  
}  
cout << '\n';
```

0 1 2 3 4 5

2 3 4 5 0 1

rotate

STL

```
vector<int> a = {0, 1, 2, 3, 4, 5};  
int n = a.size();  
for (int i=0; i<n; i++) {  
    rotate(a.begin(), a.begin()+1, a.end());  
    print(a);  
}
```

rotate

STL

```
vector<int> a = {0, 1, 2, 3, 4, 5};  
int n = a.size();  
for (int i=0; i<n; i++) {  
    //rotate(a.begin(), a.begin()+(n-1), a.end());  
    rotate(a.rbegin(), a.rbegin()+1, a.rend());  
    print(a);  
}
```

바구니 순서 바꾸기

<https://www.acmicpc.net/problem/10812>

- <https://gist.github.com/Baekjoon/7ca7893ac05d3472b31c>

swap

STL

`swap(a, b)`

a와 b에 들어있던 값을 바꾼다

swap

STL

```
int a = 10, b = 20;  
cout << a << ' ' << b << '\n';  
swap(a,b);  
cout << a << ' ' << b << '\n';  
swap(a,b);  
cout << a << ' ' << b << '\n';
```

swap

STL

```
vector<int> a = {1, 2};  
vector<int> b = {3, 4};  
cout << a[0] << ' ' << b[0] << '\n';  
swap(a,b);  
cout << a[0] << ' ' << b[0] << '\n';  
swap(a,b);  
cout << a[0] << ' ' << b[0] << '\n';
```

공 바꾸기

<https://www.acmicpc.net/problem/10813>

- <https://gist.github.com/Baekjoon/3efc6fceb418062428f>

unique

STL

28

`unique(begin, end)`

`unique`는 `[begin, end)` 구간에서 연속되는 같은 값을 하나를 제외하고 제거

실제로 컨테이너의 크기를 줄이거나 제거하지 않는다
중복을 덮어쓰우거나 시프트 시키는 방식으로 작동한다

중복을 제거한 후의 `end` 이터레이터를 리턴한다

unique

STL

```
vector<int> a = {1, 1, 2, 2, 2, 3, 1, 1, 1, 2, 2, 2, 2};  
for (int x : a) {  
    cout << x << ' '  
}  
cout << '\n';  
auto last = unique(a.begin(), a.end());  
for (int x : a) {  
    cout << x << ' '  
}  
cout << '\n';  
for (auto it = a.begin(); it != last; ++it) {  
    cout << *it << ' '  
}  
cout << '\n';
```

unique

STL

1 1 2 2 2 3 1 1 1 2 2 2 2

1 2 3 1 2 3 1 1 1 2 2 2 2

1 2 3 1 2

unique

STL

31

```
vector<int> a = {1, 1, 2, 2, 2, 3, 1, 1, 1, 2, 2, 2, 2};
```

```
auto last = unique(a.begin(), a.end());
```

```
a.erase(last, a.end());
```

```
for (int x : a) {
```

```
    cout << x << ' ';
```

```
}
```

```
cout << '\n';
```

```
1 2 3 1 2
```

unique

32

STL

```
vector<int> a = {1, 1, 2, 2, 2, 3, 1, 1, 1, 2, 2, 2, 2};  
sort(a.begin(), a.end());  
auto last = unique(a.begin(), a.end());  
a.erase(last, a.end());  
for (int x : a) {  
    cout << x << ' ';  
}  
cout << '\n';
```

1 2 3

sort

STL

33

```
sort(begin, end)
```

[begin, end)를 <를 기준으로 정렬한다.

```
sort(begin, end, cmp)
```

[begin, end)를 cmp를 기준으로 정렬한다.

sort

STL

```
vector<int> a = {5, 3, 2, 1, 4};
```

```
for (int x : a) {  
    cout << x << ' '  
}
```

```
cout << '\n';
```

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

```
for (int x : a) {  
    cout << x << ' '  
}
```

```
cout << '\n';
```

수 정렬하기

<https://www.acmicpc.net/problem/2750>

sort

STL

```
bool cmp(const int &u, const int &v) {  
    return u > v;  
}
```

```
vector<int> a = {5, 3, 2, 1, 4};
```

```
sort(a.begin(), a.end(), greater<int>());  
sort(a.begin(), a.end(), cmp);  
sort(a.begin(), a.end(), [](int u, int v) {  
    return u > v;  
});
```

단어 정렬

<https://www.acmicpc.net/problem/1181>

```
sort(a.begin(), a.end(), [](string u, string v) {  
    if (u.size() < v.size()) {  
        return true;  
    } else {  
        return false;  
    }  
});
```

단어 정렬

<https://www.acmicpc.net/problem/1181>

```
sort(a.begin(), a.end(), [](string u, string v) {  
    if (u.size() < v.size()) {  
        return true;  
    } else if (u.size() == v.size()) {  
        return u < v;  
    } else {  
        return false;  
    }  
});
```

단어 정렬

<https://www.acmicpc.net/problem/1181>

```
sort(a.begin(), a.end(), [](string u, string v) {  
    if (u.size() == v.size()) {  
        return u < v;  
    } else {  
        return u.size() < v.size();  
    }  
});
```

단어 정렬

40

<https://www.acmicpc.net/problem/1181>

```
sort(a.begin(), a.end(), [](string u, string v) {  
    return (u.size() < v.size()) || (u.size() == v.size() && u < v);  
});
```


단어 정렬

<https://www.acmicpc.net/problem/1181>

- <https://gist.github.com/Baekjoon/18a26d0cbbf4068c8e31>

좌표 정렬하기

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

- (x, y) 가 여러 개 있을 때, x 가 증가하는 순으로, 같으면 y 가 증가하는 순서로 정렬하는 문제
- `pair`를 사용하면 편한다.

```
int n;
scanf("%d", &n);
vector<pair<int, int>> a(n);
for (int i=0; i<n; i++) {
    scanf("%d %d", &a[i].first, &a[i].second);
}
sort(a.begin(), a.end());
for (int i=0; i<a.size(); i++) {
    printf("%d %d\n", a[i].first, a[i].second);
}
```

좌표 정렬하기

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

- 직접 struct를 구현하는 경우에는 비교 함수를 만들어 줘야 한다.

```
struct Point {  
    int x, y;  
}  
  
bool cmp(const Point &u, const Point &v) {  
    if (u.x < v.x) {  
        return true;  
    } else if (u.x == v.x) {  
        return u.y < v.y;  
    } else {  
        return false;  
    }  
}
```

좌표 정렬하기

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

- cmp 함수는 u가 v의 앞에 오는 것이면 true, 아니면 false 이다.
- const와 &는 붙여야 한다.

```
bool cmp(const Point &u, const Point &v) {  
    if (u.x < v.x) {  
        return true;  
    } else if (u.x == v.x) {  
        return u.y < v.y;  
    } else {  
        return false;  
    }  
}
```

좌표 정렬하기

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

- 비교 함수를 만드는 경우에는 3번째 인자로 함수 이름을 넘겨줘야 한다.

```
bool cmp(const Point &u, const Point &v) {  
    if (u.x < v.x) {  
        return true;  
    } else if (u.x == v.x) {  
        return u.y < v.y;  
    } else {  
        return false;  
    }  
}  
  
sort(a.begin(), a.end(), cmp);
```

좌표 정렬하기

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

- < 연산자를 over loading 할 수도 있다. 이 경우에는 3번째 인자가 필요 없다.

```
struct Point {  
    int x, y;  
    bool operator < (const Point &v) const {  
        if (x < v.x) {  
            return true;  
        } else if (x == v.x) {  
            return y < v.y;  
        } else {  
            return false;  
        }  
    }  
}
```

좌표 정렬하기

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

- C++ (pair): <https://gist.github.com/Baekjoon/c6c920ca4fd31eb9960e>
- C++ (struct, cmp): <https://gist.github.com/Baekjoon/f3d180e438e2087a5a97>
- C++ (struct, 연산자 오버로딩): <https://gist.github.com/Baekjoon/180de19581f0b55c433c>

좌표 정렬하기 2

48

<https://www.acmicpc.net/problem/11651>

- (x, y) 가 여러 개 있을 때, y 가 증가하는 순으로, 같으면 x 가 증가하는 순서로 정렬하는 문제

좌표 정렬하기 2

<https://www.acmicpc.net/problem/11651>

- C++ (pair, cmp): <https://gist.github.com/Baekjoon/df02385467e2c6e36d59>
- C++ (pair, 뒤집어서 저장): <https://gist.github.com/Baekjoon/7cee7721722d6789769c>

국영수

50

<https://www.acmicpc.net/problem/10825>

- 도현이네 반 학생 N명의 이름과 국어, 영어, 수학 점수가 주어진다.
 - 다음과 같은 조건으로 학생의 성적을 정렬하는 문제
1. 국어 점수가 감소하는 순서로
 2. 국어 점수가 같으면 영어 점수가 증가하는 순서로
 3. 국어 점수와 영어 점수가 같으면 수학 점수가 감소하는 순서로
 4. 모든 점수가 같으면 이름이 사전 순으로 증가하는 순서로

국영수

<https://www.acmicpc.net/problem/10825>

- C++: <https://gist.github.com/Baekjoon/112bdd3b8d53d14d0f35>
- C++ (tuple 사용): <https://gist.github.com/Baekjoon/cfbc58f340f2043213d0>

나이순 정렬

<https://www.acmicpc.net/problem/10814>

- 온라인 저지에 가입한 사람들의 나이와 이름이 가입한 순서대로 주어진다.
- 회원들을 나이가 증가하는 순으로, 나이가 같으면 먼저 가입한 사람이 앞에 오는 순서로 정렬하는 문제
- 가입한 순서는 입력으로 들어오지 않기 때문에, 따로 저장해줘야 한다.

```
struct Person {  
    int age;  
    string name;  
    int join;  
};
```

나이순 정렬

53

<https://www.acmicpc.net/problem/10814>

- <https://gist.github.com/Baekjoon/633c9eaf24cd4606ef01>

stable_sort

STL

```
stable_sort(begin, end)
```

[begin, end)를 <를 기준으로 정렬한다.

```
stable_sort(begin, end, cmp)
```

[begin, end)를 cmp를 기준으로 정렬한다.

Stable Sorting

55

Stable Sorting

- 예를 들어 다음과 같이 카드가 있는 경우를 생각해보자.
- 7♠, 5♥, 2♥, 5♠
- 위의 카드를 번호가 증가하는 순서로 정렬했을 때, 5♥와 5♠의 순서에 대해 생각해본다.
- 2♥, 5♥, 5♠, 7♠
- 와 같이 정렬이 될 수도 있고
- 2♥, 5♠, 5♥, 7♠
- 와 같이 정렬이 될 수도 있다.
- 같은 것이 있는 경우에 정렬하기 전의 순서가 유지되는 정렬 알고리즘을 Stable Sorting 알고리즘이라고 한다.

나이순 정렬

56

<https://www.acmicpc.net/problem/10814>

- <https://gist.github.com/Baekjoon/e183e261cb91790a5618>

binary_search

57

STL

`binary_search(begin, end, value)`

[begin, end)에서 value를 찾으면 true, 못 찾으면 false

`stable_sort(begin, end, value, cmp)`

[begin, end)에서 value를 cmp를 기준으로 찾으면 true, 못 찾으면 false

binary_search

58

STL

```
vector<int> a = {1, 5, 6, 7, 10, 20};
```

```
for (int i=1; i<=10; i++) {  
    cout << i << ": ";  
    cout << binary_search(a.begin(), a.end(), i) << '\n';  
}
```

숫자 외우기

<https://www.acmicpc.net/problem/10815>

- <https://gist.github.com/Baekjoon/bc5f8660300aadcd60d4>
- <https://gist.github.com/Baekjoon/6446a941d1bbb8186af6>

lower_bound/upper_bound

60

STL

`lower_bound(begin, end, value)`

`lower_bound(begin, end, value, cmp)`

`upper_bound(begin, end, value)`

`upper_bound(begin, end, value, cmp)`

`lower_bound`: `[begin, end)`에서 `value`보다 작지 않은 첫 번째 이터레이터

`upper_bound`: `[begin, end)`에서 `value`보다 큰 첫 번째 이터레이터

lower_bound/upper_bound

STL

```
vector<int> a = {1, 3, 4, 5, 7, 7, 8};
```

```
for (int i=1; i<=10; i++) {  
    auto l = lower_bound(a.begin(), a.end(), i);  
    auto r = upper_bound(a.begin(), a.end(), i);  
  
    cout << i << ": ";  
    cout << "lower_bound: " << (l-a.begin()) << ' ';  
    cout << "upper_bound: " << (r-a.begin());  
    cout << '\n';  
}
```

lower_bound/upper_bound

STL

```
1: lower_bound: 0 upper_bound: 1
2: lower_bound: 1 upper_bound: 1
3: lower_bound: 1 upper_bound: 2
4: lower_bound: 2 upper_bound: 3
5: lower_bound: 3 upper_bound: 4
6: lower_bound: 4 upper_bound: 4
7: lower_bound: 4 upper_bound: 6
8: lower_bound: 6 upper_bound: 7
9: lower_bound: 7 upper_bound: 7
10: lower_bound: 7 upper_bound: 7
```

숫자 카드 2

<https://www.acmicpc.net/problem/10815>

- <https://gist.github.com/Baekjoon/cd8471e453d9289207f7>

equal_range

STL

`equal_range(begin, end, value)`

`equal_range(begin, end, value, cmp)`

lower_bound, upper_bound 를 pair 형태로 리턴한다

숫자 카드 2

65

<https://www.acmicpc.net/problem/10815>

- <https://gist.github.com/Baekjoon/3eb1336d7c146e00824d>

min/max

STL

`min(a, b)`

`min(a, b, cmp)`

`min(initializer_list)`

`min(initializer_list, cmp)`

`max(a, b)`

`max(a, b, cmp)`

`max(initializer_list)`

`max(initializer_list, cmp)`

min/max

STL

```
cout << min(2, 3) << '\n';
```

```
cout << max(2, 3) << '\n';
```

```
int a = 10, b = 20, c = 30;
```

```
cout << min(min(a, b), c) << '\n';
```

```
cout << min({a, b, c}) << '\n';
```

```
cout << max(max(a, b), c) << '\n';
```

```
cout << max({a, b, c}) << '\n';
```

min/max

STL

```
string u = "long string";
```

```
string v = "short";
```

```
cout << min(u, v) << '\n';
```

```
cout << min(u, v, [](string u, string v) {
```

```
    return u.size() < v.size();
```

```
}) << '\n';
```

세 수

<https://www.acmicpc.net/problem/10817>

- <https://gist.github.com/Baekjoon/b71b19d4c9f206c43b3b>

minmax

70

STL

```
minmax(a, b)
```

```
minmax(a, b, cmp)
```

```
minmax(initializer_list)
```

```
minmax(initializer_list, cmp)
```

min과 max를 동시에 구할 수 있다

세 수

<https://www.acmicpc.net/problem/10817>

- <https://gist.github.com/Baekjoon/d65b66ee56b79b86d612>

min_element/max_element

STL

```
min_element(begin, end)
```

```
min_element(begin, end, cmp)
```

```
max_element(begin, end)
```

```
max_element(begin, end, cmp)
```

[begin, end)에서 최소/최대값의 이터레이터를 구한다

min_element/max_element

73

STL

```
vector<int> a = {4, 2, 1, 5, 7, 3};
```

```
auto it = min_element(a.begin(), a.end());
```

```
cout << "최소: " << *it << ", 위치: " << (it-a.begin()) << '\n';
```

```
it = max_element(a.begin(), a.end());
```

```
cout << "최대: " << *it << ", 위치: " << (it-a.begin()) << '\n';
```

최소, 최대

<https://www.acmicpc.net/problem/10818>

- <https://gist.github.com/Baekjoon/1d6fddc9dfb9b16e49dc>

minmax_element

75

STL

```
minmax_element(begin, end)
```

```
minmax_element(begin, end, cmp)
```

[begin, end)에서 최소, 최대값의 이터레이터를 구한다

최소, 최대

76

<https://www.acmicpc.net/problem/10818>

- <https://gist.github.com/Baekjoon/1159eb8cd4f7e69b2a90>

next_permutation

STL

```
next_permutation(begin, end)
```

```
next_permutation(begin, end, cmp)
```

```
prev_permutation(begin, end)
```

```
prev_permutation(begin, end, cmp)
```

[begin, end)를 순열이라고 생각했을 때, 사전 순으로 다음에 오는 순열을 만든다

마지막 순열이면 `false`를 리턴

next_permutation

STL

```
vector<int> a = {3, 1, 2};
```

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

```
do {
```

```
    for (int x: a) {
```

```
        cout << x << ' ';
```

```
    }
```

```
    cout << '\n';
```

```
} while(next_permutation(a.begin(), a.end()));
```

차이를 최대로

<https://www.acmicpc.net/problem/10819>

- <https://gist.github.com/Baekjoon/f141eea58a7313b555fd>