

유형

이분탐색, stl_unique with sort and erase

풀이

인덱스 유지가 필요한 문제 유형인데 정렬을 해야 lower bound 사용가능하다.

cpp에서 단순 할당으로 깊은 복사가 된다.

unique 함수는 연속된 중복값을 지운 부분 배열의 end()를 리턴한다.

따라서 미리 정렬하고 erase로 해당 반복자를 시작점으로 삼아 arr.end()까지 지우면 arr의 중복되지 않은 원소들의 정렬 결과를 얻을 수있다.

문제에서 중복 원소는 카운팅을 안하기 때문에 이 작업을 수행해야 한다.

그리고 개수를 구할때 lower_bound는 반복자를 리턴하므로 begin 반복자를 빼야 스칼라 값을 얻는다.

코드

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
int main(){
```

```
    cin.tie(0); ios::sync_with_stdio(0);
```

```
    int n; // 10^6
```

```
    cin>>n;
```

```
    vector<int> originalArr(n);
```

```
    for(int i=0; i<n; i++) cin>> originalArr[i];
```

```
    vector<int> sortedArr = originalArr;
```

```

sort(sortedArr.begin(), sortedArr.end());

sortedArr.erase(unique(sortedArr.begin(), sortedArr.end()), sortedArr.end()); // 중복 원
소를 제거합니다.
// unique는 연속된 중복값들을 하나로 압축. 따라서 sort와 같이 쓰여야함
// 압축 후 end()를 리턴하는데 erase로 이 반복자를 start로 해서 erase에 넣어야함

for(int i=0; i<n; i++){
    cout<<lower_bound(sortedArr.begin(), sortedArr.end(), originalArr[i]) -
sortedArr.begin()<<" ";
} // 반복자를 리턴하므로 begin()을 빼야 개수 얻음
}

```