

# 컴퓨터공학 All in One

---

C/C++ 문법, 자료구조 및 심화 프로젝트 (나동빈)  
제 27강 - 계수 정렬

# 학습 목표

## 계수 정렬

- 1) 계수 정렬의 원리를 이해하고 이를 C언어로 구현할 수 있습니다.

# 계수 정렬

## 계수 정렬

계수 정렬(Counting Sort)는 크기를 기준으로 데이터의 개수를 세는 정렬 알고리즘입니다. 각 데이터를 바로 크기를 기준으로 분류하므로  $O(N)$ 의 시간 복잡도를 가집니다.

# 계수 정렬

## 계수 정렬

계수 정렬(Counting Sort)는 크기를 기준으로 데이터의 개수를 세는 정렬 알고리즘입니다. 각 데이터를 바로 크기를 기준으로 분류하므로  $O(N)$ 의 시간 복잡도를 가집니다.

인덱스	0	1	2	3
원소	0	0	0	0

2	1	0	2	2	1	3	1	0	3
---	---	---	---	---	---	---	---	---	---

# 계수 정렬

## 계수 정렬

계수 정렬(Counting Sort)는 크기를 기준으로 데이터의 개수를 세는 정렬 알고리즘입니다. 각 데이터를 바로 크기를 기준으로 분류하므로  $O(N)$ 의 시간 복잡도를 가집니다.

인덱스	0	1	2	3
원소	0	0	1	0

2	1	0	2	2	1	3	1	0	3
---	---	---	---	---	---	---	---	---	---

# 계수 정렬

## 계수 정렬

계수 정렬(Counting Sort)는 크기를 기준으로 데이터의 개수를 세는 정렬 알고리즘입니다. 각 데이터를 바로 크기를 기준으로 분류하므로  $O(N)$ 의 시간 복잡도를 가집니다.

인덱스	0	1	2	3
원소	0	1	1	0

2	1	0	2	2	1	3	1	0	3
---	---	---	---	---	---	---	---	---	---

# 계수 정렬

## 계수 정렬

계수 정렬(Counting Sort)는 크기를 기준으로 데이터의 개수를 세는 정렬 알고리즘입니다. 각 데이터를 바로 크기를 기준으로 분류하므로  $O(N)$ 의 시간 복잡도를 가집니다.

인덱스	0	1	2	3
원소	1	1	1	0

2	1	0	2	2	1	3	1	0	3
---	---	---	---	---	---	---	---	---	---

# 계수 정렬

## 계수 정렬

계수 정렬(Counting Sort)는 크기를 기준으로 데이터의 개수를 세는 정렬 알고리즘입니다. 각 데이터를 바로 크기를 기준으로 분류하므로  $O(N)$ 의 시간 복잡도를 가집니다.

인덱스	0	1	2	3
원소	1	1	2	0

2	1	0	2	2	1	3	1	0	3
---	---	---	---	---	---	---	---	---	---



# 계수 정렬

## 계수 정렬

계수 정렬(Counting Sort)는 크기를 기준으로 데이터의 개수를 세는 정렬 알고리즘입니다. 각 데이터를 바로 크기를 기준으로 분류하므로  $O(N)$ 의 시간 복잡도를 가집니다.

인덱스	0	1	2	3
원소	1	1	3	0

2	1	0	2	2	1	3	1	0	3
---	---	---	---	---	---	---	---	---	---

# 계수 정렬

## 계수 정렬

계수 정렬(Counting Sort)는 크기를 기준으로 데이터의 개수를 세는 정렬 알고리즘입니다. 각 데이터를 바로 크기를 기준으로 분류하므로  $O(N)$ 의 시간 복잡도를 가집니다.

인덱스	0	1	2	3
원소	1	2	3	0

2	1	0	2	2	1	3	1	0	3
---	---	---	---	---	---	---	---	---	---

# 계수 정렬

## 계수 정렬

계수 정렬(Counting Sort)는 크기를 기준으로 데이터의 개수를 세는 정렬 알고리즘입니다. 각 데이터를 바로 크기를 기준으로 분류하므로  $O(N)$ 의 시간 복잡도를 가집니다.

인덱스	0	1	2	3
원소	1	2	3	1

2	1	0	2	2	1	3	1	0	3
---	---	---	---	---	---	---	---	---	---

# 계수 정렬

## 계수 정렬

계수 정렬(Counting Sort)는 크기를 기준으로 데이터의 개수를 세는 정렬 알고리즘입니다. 각 데이터를 바로 크기를 기준으로 분류하므로  $O(N)$ 의 시간 복잡도를 가집니다.

인덱스	0	1	2	3
원소	1	3	3	1

2	1	0	2	2	1	3	1	0	3
---	---	---	---	---	---	---	---	---	---

# 계수 정렬

## 계수 정렬

계수 정렬(Counting Sort)는 크기를 기준으로 데이터의 개수를 세는 정렬 알고리즘입니다. 각 데이터를 바로 크기를 기준으로 분류하므로  $O(N)$ 의 시간 복잡도를 가집니다.

인덱스	0	1	2	3
원소	2	3	3	1

2	1	0	2	2	1	3	1	0	3
---	---	---	---	---	---	---	---	---	---

# 계수 정렬

## 계수 정렬

계수 정렬(Counting Sort)는 크기를 기준으로 데이터의 개수를 세는 정렬 알고리즘입니다. 각 데이터를 바로 크기를 기준으로 분류하므로  $O(N)$ 의 시간 복잡도를 가집니다.

인덱스	0	1	2	3
원소	2	3	3	2

2	1	0	2	2	1	3	1	0	3
---	---	---	---	---	---	---	---	---	---

# 계수 정렬

## 계수 정렬

계수 정렬(Counting Sort)는 크기를 기준으로 데이터의 개수를 세는 정렬 알고리즘입니다. 각 데이터를 바로 크기를 기준으로 분류하므로  $O(N)$ 의 시간 복잡도를 가집니다.

인덱스	0	1	2	3
원소	2	3	3	2

차례대로 원소의 개수만큼 출력: 0 0 1 1 1 2 2 2 3 3

# 계수 정렬

## 계수 정렬

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#define MAX_VALUE 10001

int n, m;
int a[MAX_VALUE];

int main() {
    scanf("%d", &n);
    for (int i = 0; i < n; i++) { scanf("%d", &m); a[m]++; }
    for (int i = 0; i < MAX_VALUE; i++) {
        while (a[i] != 0) { printf("%d ", i); a[i]--; }
    }
    system("pause");
}
```



# 배운 내용 정리하기

## 계수 정렬

- 1) 계수 정렬은 시간 복잡도가  $O(N)$ 인 정렬 알고리즘입니다.
- 2) 계수 정렬은 데이터의 크기가 한정적일 때 사용할 수 있습니다.