

컴퓨터공학 All in One

C/C++ 문법, 자료구조 및 심화 프로젝트 (나동빈)
제 28강 - 기수 정렬

학습 목표

기수 정렬

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

기수 정렬

기수 정렬

기수 정렬(Radix Sort)는 자릿수를 기준으로 차례대로 데이터를 정렬하는 알고리즘입니다. 각 데이터를 자릿수를 기준으로 분류하므로 가장 큰 자릿수를 D 라고 했을 때 $O(DN)$ 의 시간 복잡도를 가집니다.

기수 정렬

기수 정렬

원소	7	84	25	341	65	30	34		
결과 배열									
자릿수 배열	0	1	2	3	4	5	6	7	8
	0	0	0	0	0	0	0	0	0

기수 정렬

기수 정렬

자릿수: 1의 자리

원소

7	84	25	341	65	30	34
---	----	----	-----	----	----	----

결과 배열

--	--	--	--	--	--	--

자릿수 배열

0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	0	0	0	0

기수 정렬

기수 정렬

자릿수: 1의 자리

원소

7	84	25	341	65	30	34
---	----	----	-----	----	----	----

결과 배열

--	--	--	--	--	--	--

자릿수 배열

0	1	2	3	4	5	6	7	8	9
1	1	0	0	2	2	0	1	0	0

기수 정렬

기수 정렬

자릿수: 1의 자리

원소

7	84	25	341	65	30	34
---	----	----	-----	----	----	----

결과 배열

--	--	--	--	--	--	--

자릿수 배열
(누적)

0	1	2	3	4	5	6	7	8	9
1	2	0	0	4	6	0	7	0	0

기수 정렬

기수 정렬

자릿수: 1의 자리

원소

7	84	25	341	65	30	34
---	----	----	-----	----	----	----

결과 배열

			34			
--	--	--	----	--	--	--

자릿수 배열
(누적)

0	1	2	3	4	5	6	7	8	9
1	2	0	0	3	6	0	7	0	0

기수 정렬

기수 정렬

자릿수: 1의 자리

원소

7	84	25	341	65	30	34
---	----	----	-----	----	----	----

결과 배열

30			34			
----	--	--	----	--	--	--

자릿수 배열
(누적)

0	1	2	3	4	5	6	7	8	9
0	2	0	0	3	6	0	7	0	0

기수 정렬

기수 정렬

자릿수: 1의 자리

원소

7	84	25	341	65	30	34
---	----	----	-----	----	----	----

결과 배열

30			34		65	
----	--	--	----	--	----	--

자릿수 배열
(누적)

0	1	2	3	4	5	6	7	8	9
0	2	0	0	3	5	0	7	0	0

기수 정렬

기수 정렬

자릿수: 1의 자리

원소

7	84	25	341	65	30	34
---	----	----	-----	----	----	----

결과 배열

30	341		34		65	
----	-----	--	----	--	----	--

자릿수 배열
(누적)

0	1	2	3	4	5	6	7	8	9
0	1	0	0	3	5	0	7	0	0

기수 정렬

기수 정렬

자릿수: 1의 자리

원소	7	84	25	341	65	30	34		
결과 배열	30	341		34	25	65			
자릿수 배열 (누적)	0	1	2	3	4	5	6	7	8
	0	1	0	0	3	4	0	7	0

기수 정렬

기수 정렬

자릿수: 1의 자리

원소

7	84	25	341	65	30	34
---	----	----	-----	----	----	----

결과 배열

30	341	84	34	25	65	
----	-----	----	----	----	----	--

자릿수 배열
(누적)

0	1	2	3	4	5	6	7	8	9
0	1	0	0	2	4	0	7	0	0

기수 정렬

기수 정렬

자릿수: 1의 자리

원소

7	84	25	341	65	30	34
---	----	----	-----	----	----	----

결과 배열

30	341	84	34	25	65	7
----	-----	----	----	----	----	---

1의자리까지 정렬 완료!

자릿수 배열
(누적)

0	1	2	3	4	5	6	7	8	9
0	1	0	0	2	4	0	6	0	0

기수 정렬

기수 정렬

자릿수: 10의 자리

원소

30	341	84	34	25	65	7
----	-----	----	----	----	----	---

결과 배열

30	341	84	34	25	65	7
----	-----	----	----	----	----	---

자릿수 배열

0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	0	0	0	0

기수 정렬

기수 정렬

자릿수: 10의 자리

원소

30	341	84	34	25	65	7
----	-----	----	----	----	----	---

결과 배열

--	--	--	--	--	--	--

자릿수 배열

0	1	2	3	4	5	6	7	8	9
1	0	1	2	1	0	1	0	1	0

기수 정렬

기수 정렬

자릿수: 10의 자리

원소

30	341	84	34	25	65	7
----	-----	----	----	----	----	---

결과 배열

--	--	--	--	--	--	--

자릿수 배열
(누적)

0	1	2	3	4	5	6	7	8	9
1	0	2	4	5	0	6	0	7	0

기수 정렬

기수 정렬

자릿수: 10의 자리

원소

30	341	84	34	25	65	7
----	-----	----	----	----	----	---

결과 배열

7						
---	--	--	--	--	--	--

자릿수 배열
(누적)

0	1	2	3	4	5	6	7	8	9
0	0	2	4	5	0	6	0	7	0

기수 정렬

기수 정렬

자릿수: 10의 자리

원소

30	341	84	34	25	65	7
----	-----	----	----	----	----	---

결과 배열

7					65	
---	--	--	--	--	----	--

자릿수 배열
(누적)

0	1	2	3	4	5	6	7	8	9
0	0	2	4	5	0	5	0	7	0

기수 정렬

기수 정렬

자릿수: 10의 자리

원소

30	341	84	34	25	65	7
----	-----	----	----	----	----	---

결과 배열

7	25				65	
---	----	--	--	--	----	--

자릿수 배열
(누적)

0	1	2	3	4	5	6	7	8	9
0	0	1	4	5	0	5	0	7	0

기수 정렬

기수 정렬

자릿수: 10의 자리

원소

30	341	84	34	25	65	7
----	-----	----	----	----	----	---

결과 배열

7	25		34		65	
---	----	--	----	--	----	--

자릿수 배열
(누적)

0	1	2	3	4	5	6	7	8	9
0	0	1	3	5	0	5	0	7	0

기수 정렬

기수 정렬

자릿수: 10의 자리

원소

30	341	84	34	25	65	7
----	-----	----	----	----	----	---

결과 배열

7	25		34		65	84
---	----	--	----	--	----	----

자릿수 배열
(누적)

0	1	2	3	4	5	6	7	8	9
0	0	1	3	5	0	5	0	6	0

기수 정렬

기수 정렬

자릿수: 10의 자리

원소

30	341	84	34	25	65	7
----	-----	----	----	----	----	---

결과 배열

7	25		34	341	65	84
---	----	--	----	-----	----	----

자릿수 배열
(누적)

0	1	2	3	4	5	6	7	8	9
0	0	1	3	4	0	5	0	6	0

기수 정렬

기수 정렬

자릿수: 10의 자리

원소

30	341	84	34	25	65	7
----	-----	----	----	----	----	---

결과 배열

7	25	30	34	341	65	84
---	----	----	----	-----	----	----

10의자리까지 정렬 완료!

자릿수 배열
(누적)

0	1	2	3	4	5	6	7	8	9
0	0	1	2	4	0	5	0	6	0

기수 정렬

기수 정렬

자릿수: 100의 자리

원소

7	25	30	34	341	65	84
---	----	----	----	-----	----	----

결과 배열

7	25	30	34	341	65	84
---	----	----	----	-----	----	----

자릿수 배열

0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	0	0	0	0

기수 정렬

기수 정렬

자릿수: 100의 자리

원소

7	25	30	34	341	65	84
---	----	----	----	-----	----	----

결과 배열

--	--	--	--	--	--	--

자릿수 배열

0	1	2	3	4	5	6	7	8	9
6	0	0	1	0	0	0	0	0	0

기수 정렬

기수 정렬

자릿수: 100의 자리

원소	7	25	30	34	341	65	84		
결과 배열									
자릿수 배열 (누적)	0	1	2	3	4	5	6	7	8
	6	0	0	7	0	0	0	0	0

기수 정렬

기수 정렬

자릿수: 100의 자리

원소

7	25	30	34	341	65	84
---	----	----	----	-----	----	----

결과 배열

					84	
--	--	--	--	--	----	--

자릿수 배열
(누적)

0	1	2	3	4	5	6	7	8	9
5	0	0	7	0	0	0	0	0	0

기수 정렬

기수 정렬

자릿수: 100의 자리

원소

7	25	30	34	341	65	84
---	----	----	----	-----	----	----

결과 배열

				65	84	
--	--	--	--	----	----	--

자릿수 배열
(누적)

0	1	2	3	4	5	6	7	8	9
4	0	0	7	0	0	0	0	0	0

기수 정렬

기수 정렬

자릿수: 100의 자리

원소

7	25	30	34	341	65	84
---	----	----	----	-----	----	----

결과 배열

				65	84	341
--	--	--	--	----	----	-----

자릿수 배열
(누적)

0	1	2	3	4	5	6	7	8	9
4	0	0	6	0	0	0	0	0	0

기수 정렬

기수 정렬

자릿수: 100의 자리

원소

7	25	30	34	341	65	84
---	----	----	----	-----	----	----

결과 배열

			34	65	84	341
--	--	--	----	----	----	-----

자릿수 배열
(누적)

0	1	2	3	4	5	6	7	8	9
3	0	0	6	0	0	0	0	0	0

기수 정렬

기수 정렬

자릿수: 100의 자리

원소

7	25	30	34	341	65	84
---	----	----	----	-----	----	----

결과 배열

		30	34	65	84	341
--	--	----	----	----	----	-----

자릿수 배열
(누적)

0	1	2	3	4	5	6	7	8	9
2	0	0	6	0	0	0	0	0	0

기수 정렬

기수 정렬

자릿수: 100의 자리

원소	7	25	30	34	341	65	84		
결과 배열		25	30	34	65	84	341		
자릿수 배열 (누적)	0	1	2	3	4	5	6	7	8
	1	0	0	6	0	0	0	0	0

기수 정렬

기수 정렬

자릿수: 100의 자리

원소

7	25	30	34	341	65	84
---	----	----	----	-----	----	----

결과 배열

7	25	30	34	65	84	341
---	----	----	----	----	----	-----

100의자리까지 정렬 완료!

자릿수 배열
(누적)

0	1	2	3	4	5	6	7	8	9
0	0	0	6	0	0	0	0	0	0

기수 정렬

기수 정렬

정렬 완료

원소

7	25	30	34	65	84	341
---	----	----	----	----	----	-----

결과 배열

7	25	30	34	65	84	341
---	----	----	----	----	----	-----

기수 정렬

기수 정렬

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#define MAX 10000

void radixSort(int *a, int n) {
    int res[MAX], maxValue = 0;
    int exp = 1;
    for (int i = 0; i < n; i++) {
        if (a[i] > maxValue) maxValue = a[i];
    }
    while (maxValue / exp > 0) { // 1의 자리부터 계산
        int bucket[10] = { 0 };
        for (int i = 0; i < n; i++) bucket[a[i] / exp % 10]++; // 자릿수 배열 처리
        for (int i = 1; i < 10; i++) bucket[i] += bucket[i - 1]; // 누적 계산
        for (int i = n - 1; i >= 0; i--) { // 같은 자릿수 끼리는 순서를 유지
            res[--bucket[a[i] / exp % 10]] = a[i];
        }
        for (int i = 0; i < n; i++) a[i] = res[i]; // 기본 배열 갱신
        exp *= 10;
    }
}
```

기수 정렬

기수 정렬

```
int main(void) {  
    int a[MAX];  
    int i, n;  
    scanf("%d", &n);  
    for (i = 0; i < n; i++) {  
        scanf("%d", &a[i]);  
    }  
    radixSort(a, n);  
    for (i = 0; i < n; i++) {  
        printf("%d ", a[i]);  
    }  
    system("pause");  
}
```

배운 내용 정리하기

기수 정렬

- 1) 기수 정렬은 시간 복잡도가 $O(DN)$ 인 정렬 알고리즘입니다.
- 2) 기수 정렬은 계수 정렬보다 약간 더 느리지만, 숫자가 매우 큰 상황에서도 사용할 수 있습니다.