

Counting Sort (계수 정렬)

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

- 1) 계수정렬은 시간복잡도가 $O(N)$ 인 정렬 알고리즘입니다.
- 2) 계수정렬은 데이터의 크기가 한정적일 때 사용할 수 있습니다.

<Before>

인덱스	0	1	2	3
원소	0	0	1	0

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

<After>

인덱스	0	1	2	3
원소	2	3	3	2

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

<Result>

차례대로 원소의 개수만큼 출력: 0 0 1 1 1 2 2 2 3 3

```

#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");
    return 0
}

```

2. SRC ver2

```

#define _CRT_SECURE_NO_WARNINGS

```

```
#include <stdio.h>
#include <stdlib.h>
#include <vcruntime_string.h>

int isFindMaxValue(int* pList)
{
    int nSize = _msize(pList) / sizeof(int);

    int nRetVal = 0;
    for (int i = 0; i < nSize; ++i)
    {
        if (nRetVal <= *(pList + i))
            nRetVal = *(pList + i);
    }

    return nRetVal;
}

int main()
{
    int nSize = 0;
    printf("size : ");
    scanf("%d", &nSize);

    int *pInputList = (int*)malloc(sizeof(int)*nSize);
    for (int i = 0; i < nSize; ++i)
    {
        printf("[%d] : ", i + 1);
        scanf("%d", pInputList + i);
    }
}
```

```

    int nMaxVal = isFindMaxValue(pInputList) + 1;
    int *pResultList = (int*)malloc(sizeof(int)*nMaxVal);
    memset(pResultList, 0, sizeof(int)*nMaxVal);

    printf("-----\n");
    printf("Before\n");

    int nIDX = 0;
    for (int i = 0; i < nSize; ++i)
    {
        printf("[%d] : %d\n", i, *(pInputList + i));
        nIDX = *(pInputList + i);

        (*(pResultList + nIDX))++;
    }

    printf("-----\n");
    printf("After\n");
    int nCNT = 0;
    for (int i = 0; i < nMaxVal; ++i)
    {
        while (*(pResultList + i) != 0)
        {
            printf("[%d] : %d\n", nCNT++, i);
            (*(pResultList + i))--;
        }
    }

```

```
system("pause");  
return 0;
```

```
}
```
