**hw7\_1.cpp**

**1. Stable sorting result**

The stable sorting result of the following input would be

Sorted data

3 10

8 10

4 20

2 25

7 25

1 30

6 30

5 80

Id is in ascending order when score is same.

**2. Code Explanation**

1) void selection\_sort\_stable(data\* list, int n)

:selection sort function with stable result

-list: data we will sort

-n: size of list

-least: index of element which has minimum data and smallest id among minimum datas

->in for loop

if (list->score[j] < list->score[least]) least = j;

// when j th score data is smaller than least th score: replace 'least' into j

if (list->score[j] == list->score[least] && list->id[j] < list->id[least]) least = j;

// when both score is same, then sort them with id

**3. Result**

**텍스트이(가) 표시된 사진

자동 생성된 설명**

**hw7\_2.cpp**

**1. Code Explanation**

1) #define BUCKETS 64, #define DIGITS 4, #define n 1000

-BUCKETS: range of each digit

-DIGITS: number of digits

-n: number of input data

2) std::random\_device rng;

std::uniform\_int\_distribution<int> random\_input(0, 16777215);

for (size\_t i = 1; i <= 1000; ++i){

input[i] = random\_input(rng);

}

-> generate 1000 random integer of range [0, 2^24-1] (1677215=2^24-1)

3) int input[1001] , int output[1001]

-input: save random input data

-output: save sorted output data

4) void RadixSort(int\* A, int d)

: divide each data of array A into 4 digits and sort datas at each digit

-int\* A: array we which will sort

-int d: number of digits

-int C[64]: array for cumulative histogram

In for loop

① Initialize array C

for (int j = 0;j < BUCKETS;j++) {

C[j] = 0;

}

② create histogram in C

for (int j = 1;j <= n;j++) {

C[(A[j] / factor) % BUCKETS] += 1;

}

③ create cumulative histogram

for (int j = 1;j < BUCKETS;j++) {

C[j] = C[j] + C[j - 1];

}

④ generate sorted output data

for (int j = n;j >= 1;j--) {

output[C[(A[j] / factor) % BUCKETS]] = A[j];

C[(A[j] / factor) % BUCKETS] -= 1;

}

//[(A[j] / factor) % BUCKETS] -> ith digit of A[j]

⑤ process next digit

factor \*= BUCKETS

**2. Result**

1000 sorted data

**텍스트이(가) 표시된 사진

자동 생성된 설명**