

Technical Report

1829008 김민영

자료구조 과제7

HW7-1 Selection Sort : Selection Sort stable

1. basic ideas and how you applied it into your code, analysis of your code(complexity)

Selection Sort 는 왼쪽리스트와 오른쪽리스트로 나누어 정렬하는 방법이다. 왼쪽은 정렬된 데이터이고 오른쪽은 아직 정렬되지 않은 데이터이다. 그 과정은 다음과 같다.

- 1) 오른쪽 리스트에서 최소값을 선택해 오른쪽 리스트의 첫번째값과 바꾼다.
- 2) 왼쪽 리스트의 크기를 증가시킨다.
- 3) 오른쪽 리스트의 크기를 감소시킨다. 오른쪽 리스트가 빌 때까지 이 세과정을 반복적으로 수행한다.

Time Complexity : Best – $O(n^2)$ Average – $O(n^2)$ Worst – $O(n^2)$

그러나, 기존 강의자료에 있는 기존 강의자료에 있는 Selection sort 는 unstable 하다. 따라서 stable 하게 고쳐주려면 각각의 데이터에 id 를 지정해주고 값을 같은데 id 는 더 작은게 있으면 작은걸 넣어주게 고쳐주면 된다.

2. result image

1) <unstable> 기존 코드

```
Input data
1      30
2      25
3      10
4      20
5      80
6      30
7      25
8      10

Sorted data
3      10
8      10
4      20
7      25
2      25
6      30
1      30
5      80
```

2) <stable> 수정한 코드

```
Input data
1      30
2      25
3      10
4      20
5      80
6      30
7      25
8      10

Sorted data
3      10
8      10
4      20
2      25
7      25
1      30
6      30
5      80
```

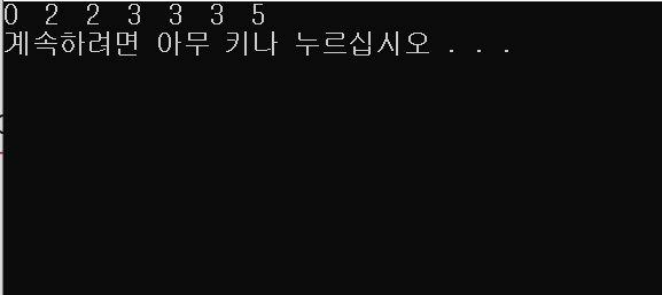
HW7-2 : Counting Sort

1. basic ideas and how you applied it into your code, analysis of your code(complexity)

Counting Sort 는 각 숫자가 나오는 횟수를 세어 히스토그램을 만들고 누적히스토그램을 만들어 정렬하는 좋은 정렬방법이다. 시간복잡도는 $O(n+k)$ 이다.

기존의 강의자료에서는 마지막 for 문에서 $j=n-1$ downto 1 로 구현하였다. 과제에서는 이를 $j=1$ to $n-1$ 로 반대로 숫자를 집어넣는 방식이었다. 이렇게 수정하여도 코드가 잘 돌아갔다. 다만 찬찬히 그림을 그리고 생각을 해보면 unstable 하다는 것을 알 수 있었다.

2. result image



```
0 2 2 3 3 3 5
계속하려면 아무 키나 누르십시오 . . .
```

HW3-3 : Radix Sort using Counting Sort

1. basic ideas and how you applied it into your code, analysis of your code(complexity)

Radix Sort 는 각 digit 에 대해 정렬하는 방법이다. Time Complexity 는 $O(d(n+k))$ 이다.

기존 강의자료에 있는 Radix Sort 는 큐를 이용하여 구현했다. 그러나 이번 과제에서는 큐를 쓰지 않고, Counting 정렬을 반복적으로 호출하여 Radix Sort 를 구현하였다.

1000 개의 24bit positive integer random number 를 생성하여 input 데이터를 만들었다.

2. result image

```
28
28
41
53
58
123
140
142
148
153
193
235
288
292
300
303
335
467
481
491
503
608
610
690
748
750
778
788
875
900
```