

Day12 - Sorting Algorithm (Counting Sort)

1. 병합 정렬 (Merge Sort)

a. 배열의 데이터가 정수 혹은 문자

2. 위상 정렬 (Topological Sort)

3. 계수 정렬 (Counting Sort)

4. 퀵 정렬 (Quick Sort)

5. 삽입 정렬 (Insertion Sort)

6. 선택 정렬 (Selection Sort)

계수 정렬 (Counting Sort)

- 특정한 조건이 부합될 때 사용할 수 있다. 단. 매우 빠른 정렬 알고리즘.
- 데이터의 개수와 데이터 중 가장 큰 최대값.
- 1,000,000 을 넘지 않을 때 가장 효과적으로 사용할 수 있다.

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

가장 큰 데이터와 가장 작은 데이터의 범위를 지정하는 리스트를 선언

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

개수 정렬 0 0 1 1 2 2 3 4 5 5 6 7 8 9 9

코드로 변경:

Counting Sort Algorithm.ipynb

```
1 array = [7, 5, 9, 0, 3, 1, 6, 2, 9, 1, 4, 8, 0, 5, 2]
2
3 count = [0]*(max(array)+1) # 모든 데이터 범위를 포함하는 리스트 선언
4
5 for i in range(len(array)):
6     count[array[i]] += 1 # 각 데이터에 해당하는 인덱스 값 증가
7
8 for i in range(len(count)): # 리스트에 기록된 정렬 정보 확인
9     for j in range(count[i]):
10         print(i, end=" ")
11
```

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