Algorithm
Counting Sort

Bill Kim(김정훈) | ibillkim@gmail.com

목차

Counting Sort

Counting Sort(계수 정렬)는 기수(Radix)정렬과 마찬가지로 각 요소를 비교하지 않고 정렬하는 정렬 알고리즘입니다.

원소간을 직접 비교하지 않고 각 원소가 몇 개 등장하는지 갯수를 세서 정렬하는 방법입니다.

해당 정렬을 위해서는 모든 원소는 양의 정수여야 하는 큰 단점이 있습니다.

정렬에 드는 시간 복잡도는 O(n+k)로서 퀵정렬이나 병합 정렬보다 일반적으로 빠릅니다.

하지만 k가 n보다 작다면 O(n)의 속도를 가지지만 k가 n보다 매우 클 경우로 O(무한)일 될 수도 있는 단점을 가지고 있습니다.

또한 계수 정렬은 다른 정렬에 비해서 많은 메모리가 사용됩니다.

Concept

기본적인 알고리즘의 컨셉을 살펴보면 아래와 같습니다.

1. 입력 배열에서 가장 큰 수를 추출한다.
2. 배열 내의 원소 값들의 갯수를 저장하는 카운팅 배열을 큰 수의 갯수만큼 만든다.
3. 카운팅 배열을 0으로 초기화한다.
4. 입력 배열을 돌면서 입력 배열의 수를 인덱스로 하여 카운팅 배열의 값을 증가시킨다.
5. 카운팅 배열의 요소들에 대해서 직전 요소들의 값을 더하여 카운팅 배열을 재조정한다.
6. 입력 배열과 동일한 크기의 출력(정렬) 배열을 만든다.
7. 입력 배열의 요소를 순서로 돌면서 입력 배열의 요소가 카운팅 배열의 인덱스를 가리키며 카운트 배열의 카운트 숫자를 하나 빼준다.
8. 카운트 배열의 숫자가 최종 출력 배열의 인덱스를 나타내며 해당 인덱스에 입력 배열의 값을 넣어준다.
9. 모든 입력 배열만큼 반복문을 돌면 모든 과정을 마친다.

Concept

만약 아래와 같은 배열이 있다고 가정합니다.

입력 배열 : [3, 2, 4, 1]

해당 입력 배열에 해당하는 카운팅 배열을 만들면 아래와 같습니다.

카운팅 배열 : [0, 1, 1, 1, 1]

카운팅 배열에서 앞 전의 값을 증가시킨 형태로 변형합니다.

누적된 카운팅 배열 : [0, 1, 2, 3, 4]

Concept

이제 카운팅 배열을 참조하여 입력된 배열의 값을 최종 출력 배열에 담습니다.

입력 배열 순서대로 데이터를 담으면 아래와 같은 과정을 거치게 됩니다.

1) 출력 배열의 2 번째 인덱스에 입력 배열 3 값을 넣음
   [- - 3 -]
2) 출력 배열의 1 번째 인덱스에 입력 배열 2 값을 넣음
   [- 2 3 -]
3) 출력 배열의 3 번째 인덱스에 입력 배열 4 값을 넣음
   [- 2 3 4]
4) 출력 배열의 0 번째 인덱스에 입력 배열 1 값을 넣음
   [1 2 3 4]

위와 같이 입력 배열에서 순서대로 최종 출력 배열로 옮기고 나면 아래와 같은 최종 정렬된 배열을 얻을 수 있습니다.

최종 정렬 배열 : [1, 2, 3, 4]

Features

Counting Sort(계수 정렬)는 아래와 같은 특징을 가진 알고리즘입니다.

1. 데이터 비교없이 원소들의 갯수를 통하여 정렬하는 알고리즘
2. 시간 복잡도가 O(n)을 갖는 엄청난 속도의 알고리즘
3. 동일한 값을 가지는 요소들을 정렬 후에도 정렬 전과 같은 순서를 가질 수 있는 안정된 정렬 알고리즘
4. 입력된 요소들의 최대 수만큼 별도의 공간이 필요하며 경우에 따라서는 매우 비효율적인 공간 낭비가 이루어질 수 있음
5. 음수가 아닌 정수의 값만 비교할 수 있기 때문에 사용이 제한적

Implementation

Swift를 활용하여 계수 정렬 알고리즘을 살펴보겠습니다.

```
func countingSort(_ array: [Int]) -> [Int] {
    guard array.count > 0 else { return [] }

    // Step 1
    // Create an array to store the count of each element
    let maxElement = array.max() ?? 0
    var countArray = [Int](repeating: 0, count: Int(maxElement + 1))

    for element in array {
        countArray[element] += 1
    }

    print("inputArray : \(array)")
    print("countArray : \(countArray)")

    // Step 2
    // Set each value to be the sum of the previous two values
    for index in 1 ..< countArray.count {
    let sum = countArray[index] + countArray[index - 1]
        countArray[index] = sum
    }

    print("countArray(accumulated) : \(countArray)”)

    ....
```

Implementation

```
    ....

    // Step 3
    // Place the element in the final array as per the number of elements before
it

    print("inputArray : \(array)")

    var sortedArray = [Int](repeating: 0, count: array.count)
    for element in array {
        countArray[element] -= 1
        sortedArray[countArray[element]] = element

        print("출력 배열의 \(countArray[element]) 번째 인덱스에 입력 배열 \(element) 값을
넣음")
        //print("countArray[\(element)] : \(countArray[element])")
    }

    print("sortedArray : \(sortedArray)")

    return sortedArray
}
```

Implementation

```
var array = [3, 2, 4, 1]
print(countingSort(array))

// inputArray : [3, 2, 4, 1]
// countArray : [0, 1, 1, 1, 1]
// countArray(accumulated) : [0, 1, 2, 3, 4]
// inputArray : [3, 2, 4, 1]
// 출력 배열의 2 번째 인덱스에 입력 배열 3 값을 넣음
// 출력 배열의 1 번째 인덱스에 입력 배열 2 값을 넣음
// 출력 배열의 3 번째 인덱스에 입력 배열 4 값을 넣음
// 출력 배열의 0 번째 인덱스에 입력 배열 1 값을 넣음
// sortedArray : [1, 2, 3, 4]
// [1, 2, 3, 4]
```

References

[1] Counting Sort : 계수 정렬 : https://bowbowbow.tistory.com/8

[2] 계수 정렬(counting sort) : https://www.zerocho.com/category/Algorithm/post/58006da88475ed00152d6c4b

[3] 계수정렬 && 기수정렬 : https://velog.io/@pa324/계수정렬-12k1akfhrm

[4] Counting Sort - 계수정렬 : https://yaboong.github.io/algorithms/2018/03/20/counting-sort/

[5] 11. 계수 정렬(Counting Sort) : https://blog.naver.com/ndb796/221228361368

References

[6] 12강 - 계수 정렬(Counting Sort) [실전 알고리즘 강좌 (Algorithm Programming Tutorial) #12] : http://www.ts-parfum.ru/video/n4kbFRn2z9M

[7] 카운팅 정렬, 래딕스 정렬 : https://ratsgo.github.io/data%20structure&algorithm/2017/10/16/countingsort/

[8] 카운팅 정렬 (Counting Sort) : https://soobarkbar.tistory.com/101

[9] Counting Sort (계수 정렬) : https://starkying.tistory.com/entry/Counting-Sort-계수-정렬

[10] 계수정렬 Counting sort : https://hongku.tistory.com/155

Thank you!