

Algorithm

Radix Sort

Bill Kim(김정훈) | ibillkim@gmail.com

목차

Radix Sort

Concept

Features

Implementation

References

Radix Sort

Radix Sort(기수 정렬)는 각 요소를 비교하지 않고 정렬하는 특별한 정렬 알고리즘입니다.

각 자리수에 해당하는 버킷을 준비하고 1의 자리부터 제일 큰 수의 자리수까지 넣다 뺀다를 반복하여 정렬하는 알고리즘입니다.

시간 복잡도가 최악과 최고 모두 $O(n)$ 이라는 말도 안되는 속도를 가지고 있지만 많은 양의 저장 공간을 차지한다는 단점이 있습니다.

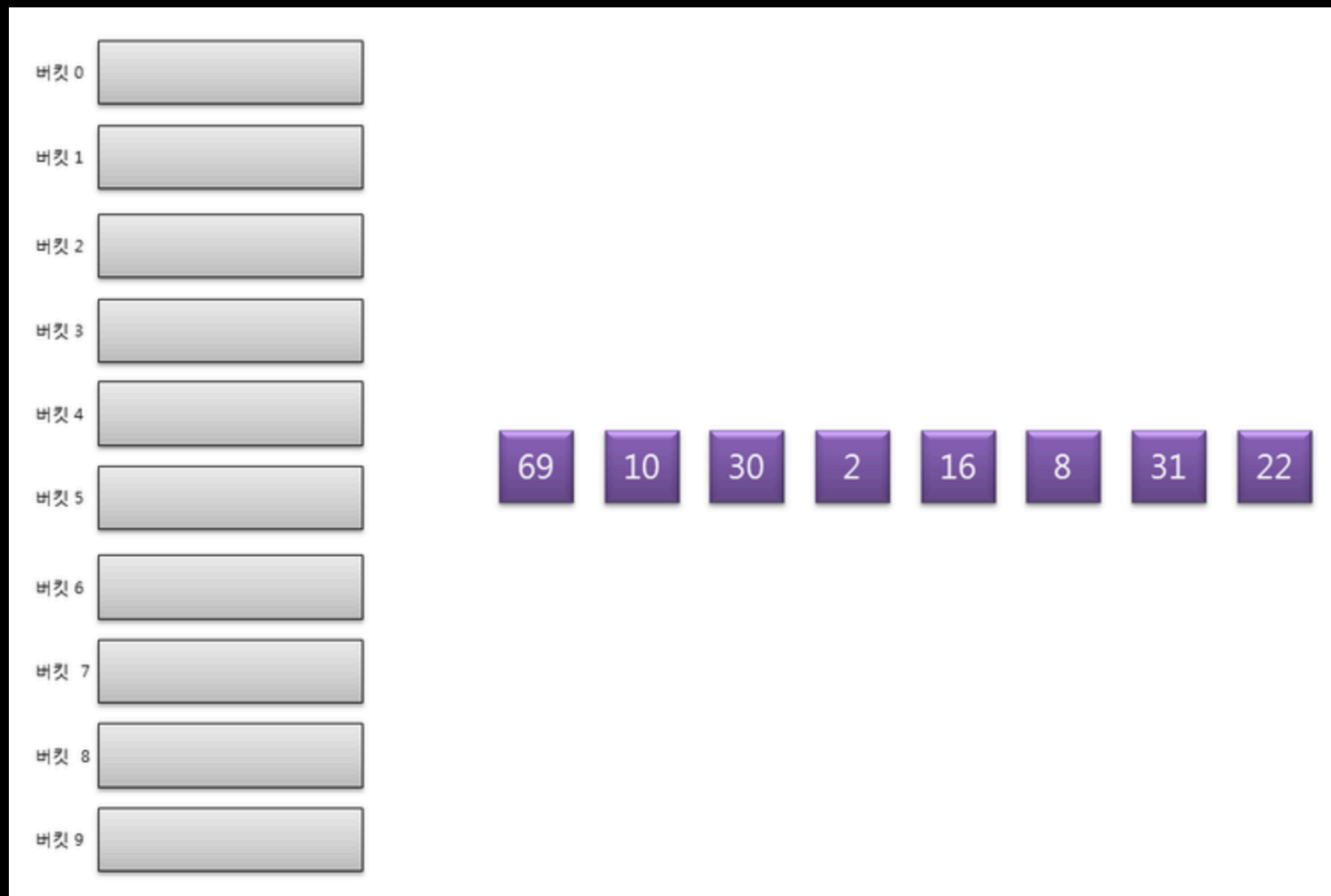
Concept

기본적인 알고리즘의 컨셉을 살펴보면 아래와 같습니다.

1. 배열 요소가 숫자일 경우 0부터 9까지의 자리수만큼의 버킷을 준비합니다.
2. 최초에는 1의 자리수를 비교하고 1의 자리수와 동일한 번호의 버킷에 요소들을 넣습니다.
3. 다음은 10의 자리수를 2번과 같은 방법으로 버킷에 넣습니다.
4. 배열 요소가 가지는 최대 자리 수 만큼 버킷에 넣다 빼면 최종 정렬된 배열 리스트를 얻을 수 있습니다.

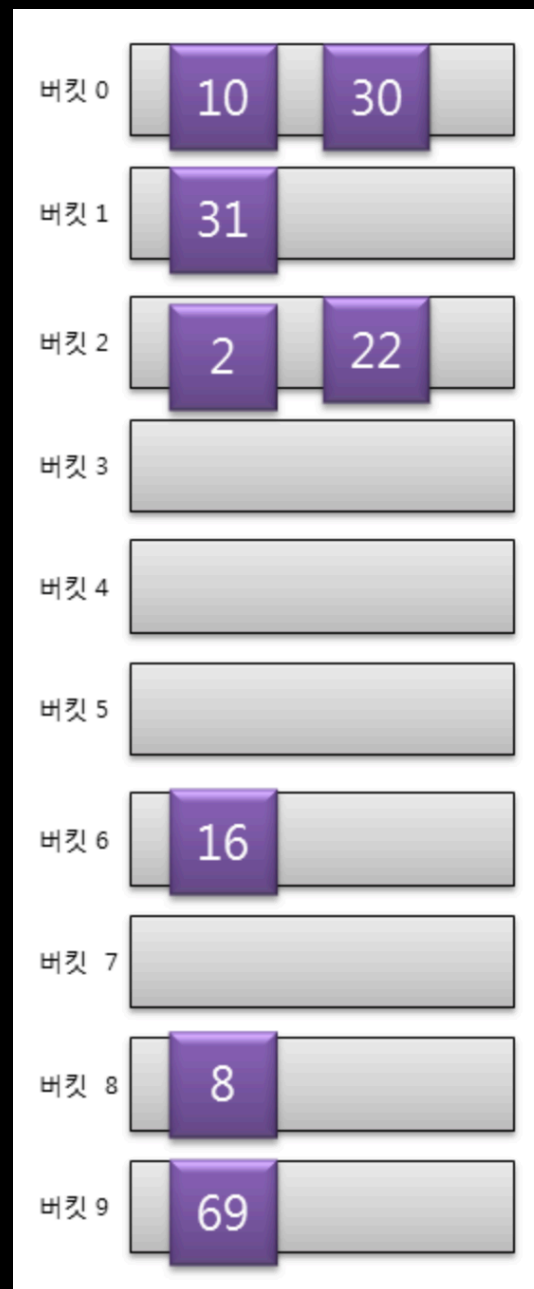
Concept

만약 아래와 같은 수가 있다고 가정합니다.



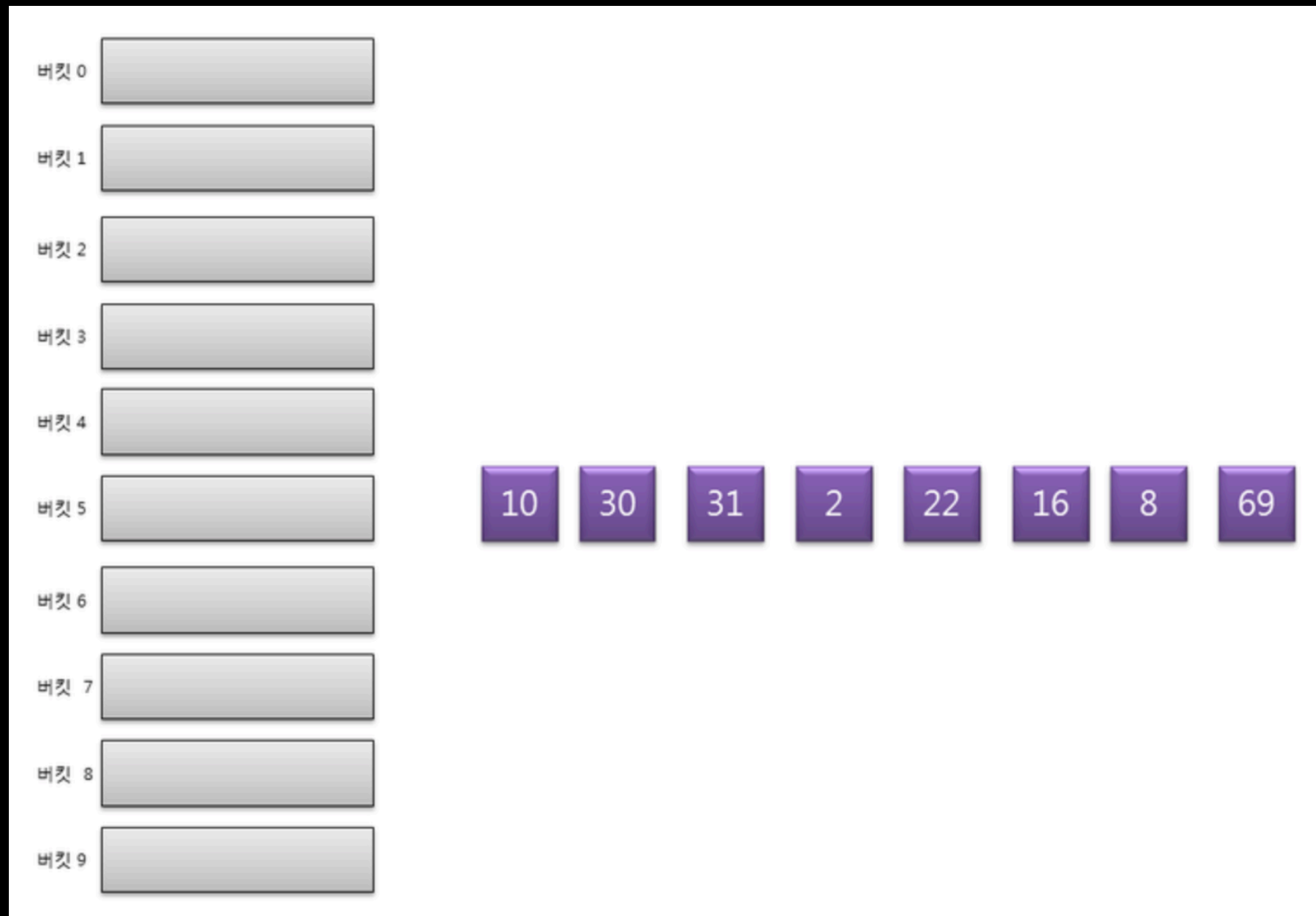
Concept

1의 자리수를 비교하여 같은 수의 버킷에 담으면 아래와 같습니다.



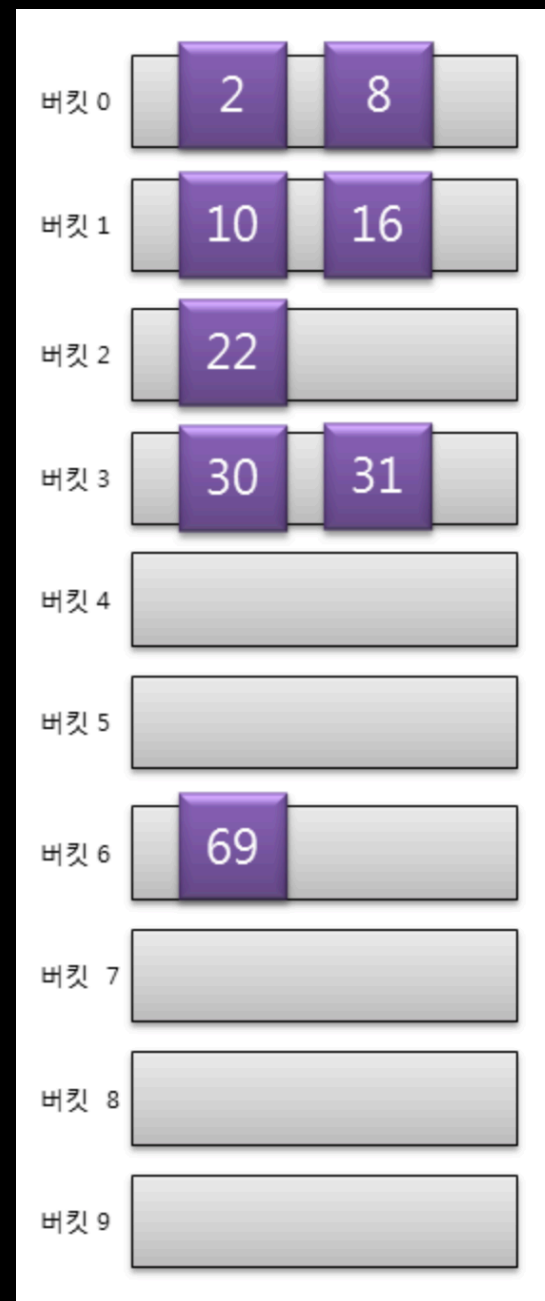
Concept

1의 자리수를 비교 후 최종적으로 아래와 같은 리스트를 얻을 수 있습니다.



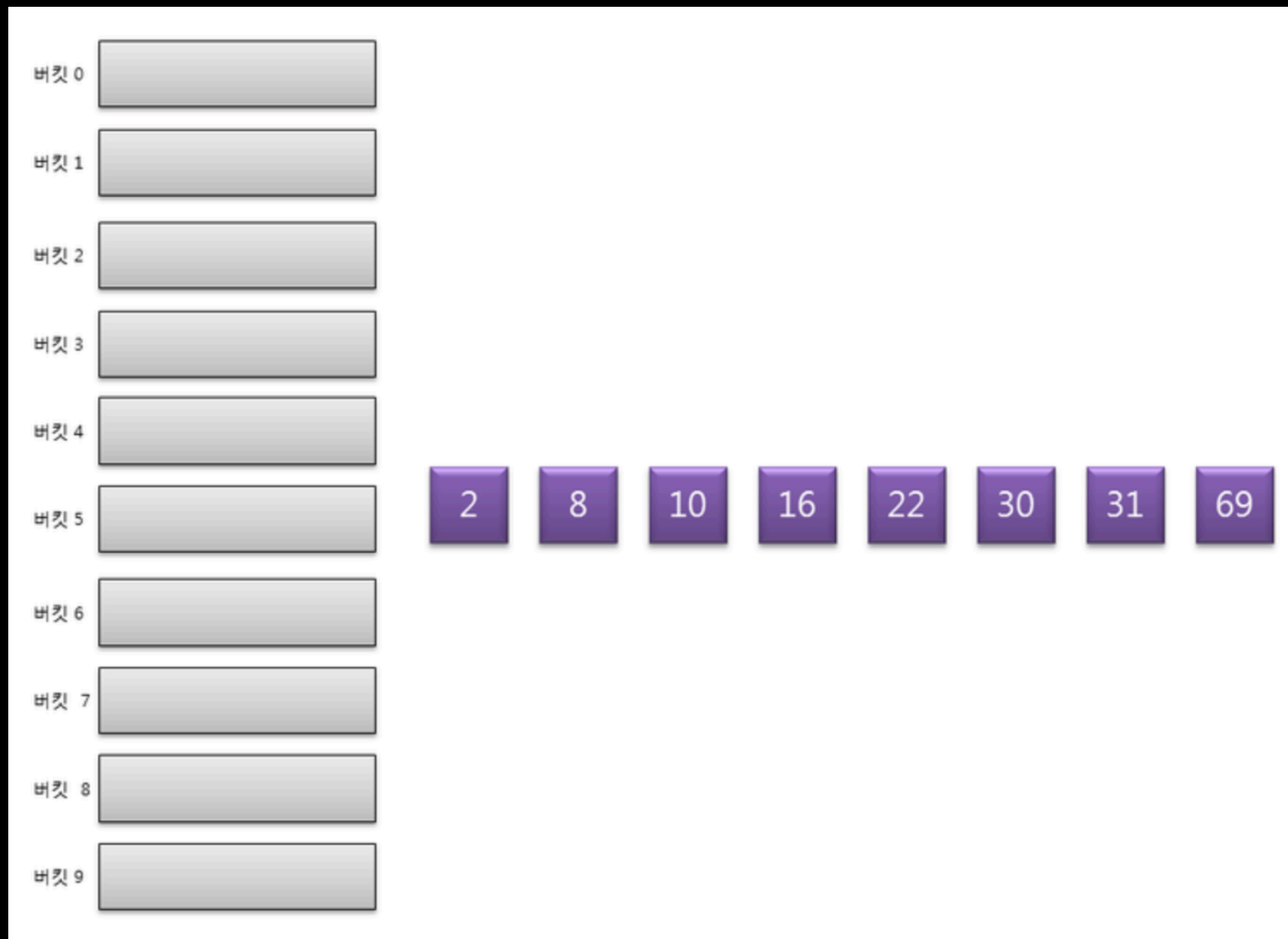
Concept

이번에는 10의 자리수에 맞는 버킷에 요소를 넣으면 아래와 같이 됩니다.



Concept

모든 배열의 요소가 정렬이 완료되었습니다.



Features

Radix Sort(기수 정렬)는 아래와 같은 특징을 가진 알고리즘입니다.

1. 데이터 비교없이 자릿수 비교를 통하여 정렬하는 알고리즘
2. 숫자일 경우 0 ~ 9까지의 자리수 비교를 위한 버킷을 준비
영문자의 경우는 a ~ z까지의 버킷 준비 필요
3. 시간 복잡도가 $O(n)$ 을 갖는 엄청난 속도의 알고리즘
4. 정렬을 위한 버킷 생성으로 인하여 많은 공간이 필요
5. 버킷에 데이터를 담고 빼는 과정으로 인한 추가 시간이 필요할 수 있음

Implementation

Swift를 활용하여 기수 정렬 알고리즘을 살펴보겠습니다.

```
func radixSort(_ array: inout [Int]) -> [Int] {  
    let radix = 10 // 0 ~ 9 까지의 10개의 자릿수  
    var done = false  
    var index: Int  
    var digit = 1  
  
    while !done {  
        done = true  
        var buckets: [[Int]] = [] // 10개의 버킷을 2중 배열로 선언  
        for _ in 1...radix { buckets.append([]) }  
  
        for number in array {  
            index = number / digit // 해당 요소의 자릿수의 숫자를 추출  
            buckets[index % radix].append(number) // 해당 자릿수 버킷에 데이터 추가  
            if done && index > 0 {  
                done = false  
            }  
        }  
  
        var i = 0  
  
        for j in 0..  
radix {  
            let bucket = buckets[j]  
            for number in bucket {  
                array[i] = number  
                i += 1  
            }  
        }  
  
        digit *= radix // 다음 자릿수 비교  
    }  
  
    return array  
}
```

Implementation

```
var array = [ 451, 5, 13, 2, 25, 131, 7, 17, 20, 8, 4, 123]  
print(radixSort(&array)) // [2, 4, 5, 7, 8, 13, 17, 20, 25, 123, 131, 451]
```

References

[1] 알고리즘) Shell/Radix Sort (셸/기수 정렬) : <https://os94.tistory.com/80?category=8147047>

[2] Radix and Shell sort : <https://www.slideshare.net/hannatamayao/radix-and-shell-sort-84534274>

[3] 기수 정렬(Radix Sort) : <https://parkdream.tistory.com/115>

[4] 06 정렬 알고리즘 - 기수 정렬(Radix Sort) : <https://lktprogrammer.tistory.com/48>

[5] [Sort] 기수 정렬(Radix Sort) : <https://palpit.tistory.com/129>

References

[6] 13.0 기수 정렬(Radix Sort) - 시작 : <https://www.youtube.com/watch?v=e8Vg0Vlrxdg>

[7] Radix Sort : <https://www.geeksforgeeks.org/radix-sort/>

[8] [정렬] 기수 정렬 (Radix Sort) (C++) : <https://yabmoons.tistory.com/248>

[9] Radix Sort(기수 정렬) : <https://sexycoder.tistory.com/74l>

[10] Radix Sort, 기수 정렬이란? : <https://dojinkimm.github.io/algorithm/2019/09/24/sort-algorithm-9.html>

Thank you!