

병합정렬

✓ 시간복잡도

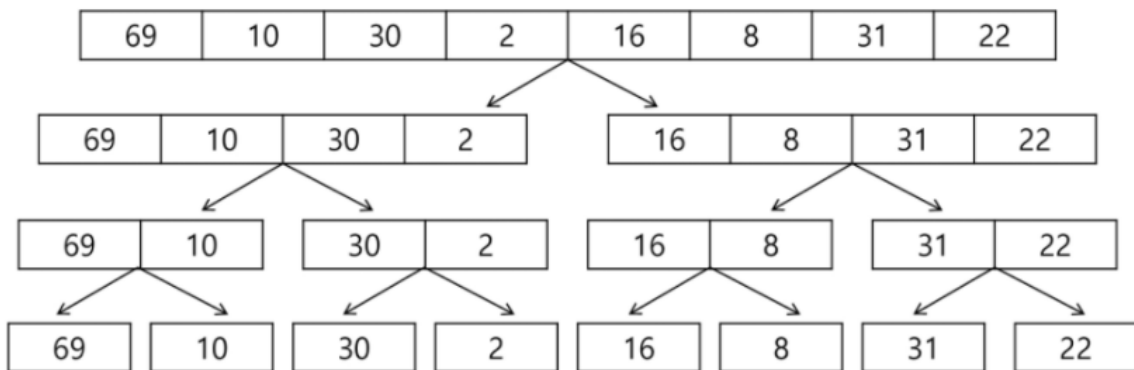
- 최선, 최악, 평균 $O(n \log n)$

✓ 개요

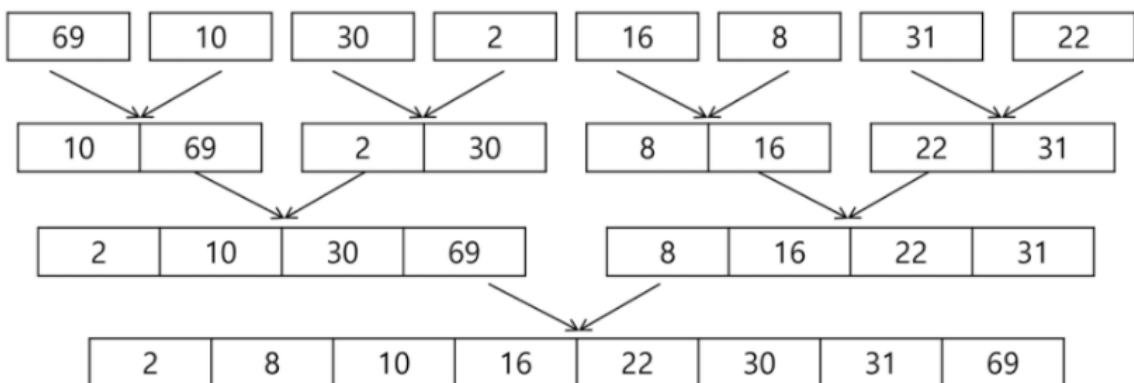
- 여러개의 정렬된 자료의 집합을 병합하여 한개의 정렬된 집합으로 만드는 방식이다.
- 분할 정복 알고리즘을 활용하여 자료를 최소단위로 나눈 후에 차례 대로 정렬하여 최종 결과를 얻어내는 Top-Down 방식이다.
- 시간복잡도는 반으로 나누는 것 $O(\log n)$, 정렬하는 것 $O(n)$ 으로 둘을 곱한 $O(n \log n)$ 의 시간 복잡도를 가진다.

✓ Preview

- 분할한다.



- 병합(정렬) 한다.



✓ Code

- 분할 과정

```
def merge_sort(m):
    if len(m) == 1:
        return m

    middle = len(m) // 2
    left = merge_sort(m[:middle])
    right = merge_sort(m[middle:])

    return merge(left, right)
```

- 병합 과정(파이썬에서는 비교하여 팝으로 넣는것도 가능)

```
def merge(left, right):
    result = []
    l, r = 0, 0
    while l < len(left) or r < len(right): # 미정렬 원소가 하나라도 남은 경우
        if l < len(left) and r < len(right): # 둘다 미정렬 원소가 있는 경우
            if left[l] <= right[r]:
                result.append(left[l])
                l += 1
            else:
                result.append(right[r])
                r += 1
        elif l < len(left): # 왼쪽만 남은 경우(오른쪽은 인덱스 끝)
            result.append(left[l])
            l += 1
        elif r < len(right): # 오른쪽만 남은 경우(왼쪽은 인덱스 끝)
            result.append(right[r])
            r += 1

    return result
```