

# Merge Sort

# Merge Sort는 가장 효율적인 알고리즘 중 하나

## 정렬 알고리즘 시간복잡도 비교

Name	Best	Avg	Worst	Run-time(정수 60,000개) 단위: sec
삽입정렬	$n$	$n^2$	$n^2$	7.438
선택정렬	$n^2$	$n^2$	$n^2$	10.842
버블정렬	$n^2$	$n^2$	$n^2$	22.894
셸 정렬	$n$	$n^{1.5}$	$n^2$	0.056
퀵 정렬	$n \log_2 n$	$n \log_2 n$	$n^2$	0.014
힙 정렬	$n \log_2 n$	$n \log_2 n$	$n \log_2 n$	0.034
병합정렬	$n \log_2 n$	$n \log_2 n$	$n \log_2 n$	0.026

# 안정 정렬에 속하며, 분할 정복 알고리즘의 하나

분할 정복 알고리즘?

문제를 작은 2개의 문제로 분리하고 각각을 해결한 다음, 결과를 모아서 원래의 문제를 해결하는 전략

대개 순환 호출을 이용한다.

안정 정렬?

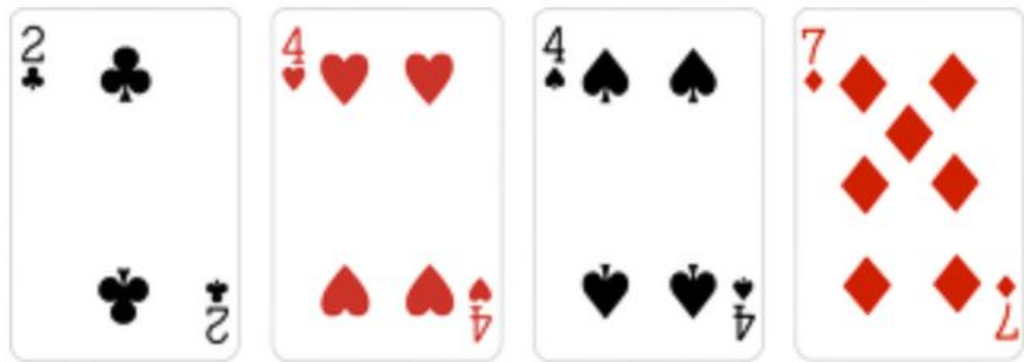
<https://godgod732.tistory.com/10>



## 안정 정렬(Stable Sort)

안정 정렬의 경우에는 정렬 후에도 원래의 순서가 유지되며, 결과는 다음과 같습니다.

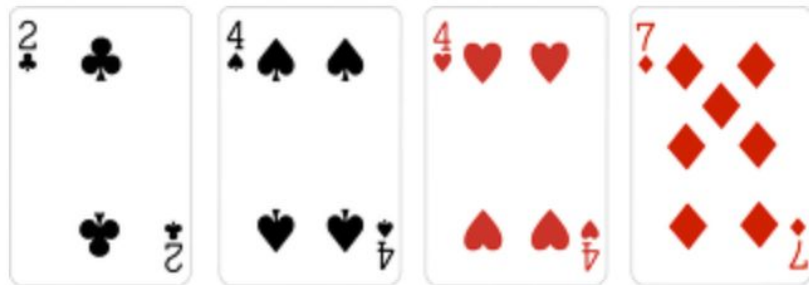
정렬된 결과에서 하트4와 스페이스4의 순서가 그대로 유지되고 있음을 확인 할 수 있습니다.



## 불안정 정렬(Unstable Sort)

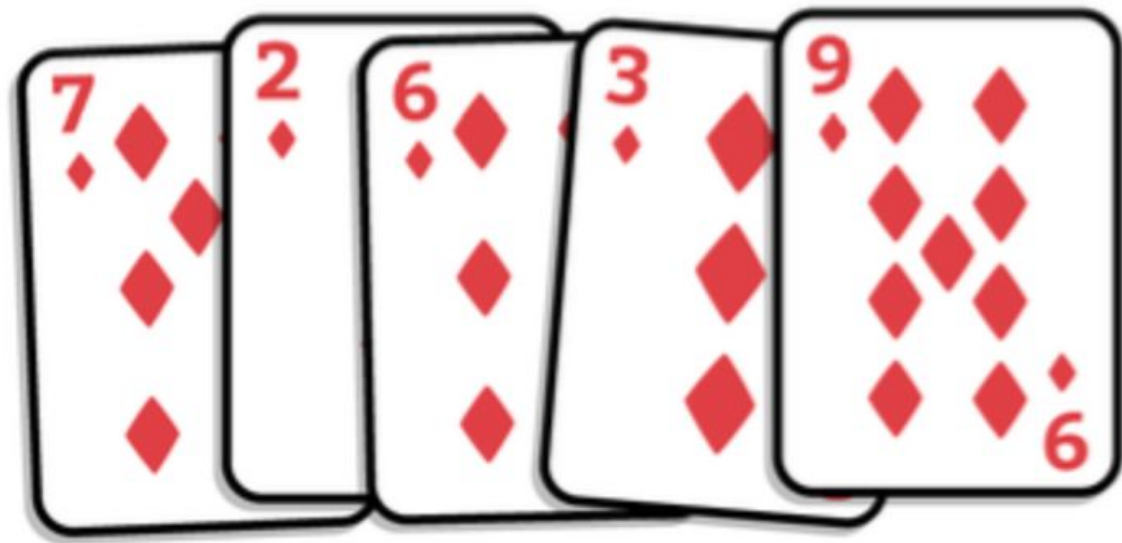
불안정 정렬의 경우에는 정렬 후에도 원래의 순서가 유지된다는 보장을 할 수 없으며, 결과는 다음과 같을 수 있습니다.

안정 정렬의 결과와는 달리 스페이스4와 하트4의 순서가 바뀐 모습을 확인 할 수 있습니다.



Merge sort 예시

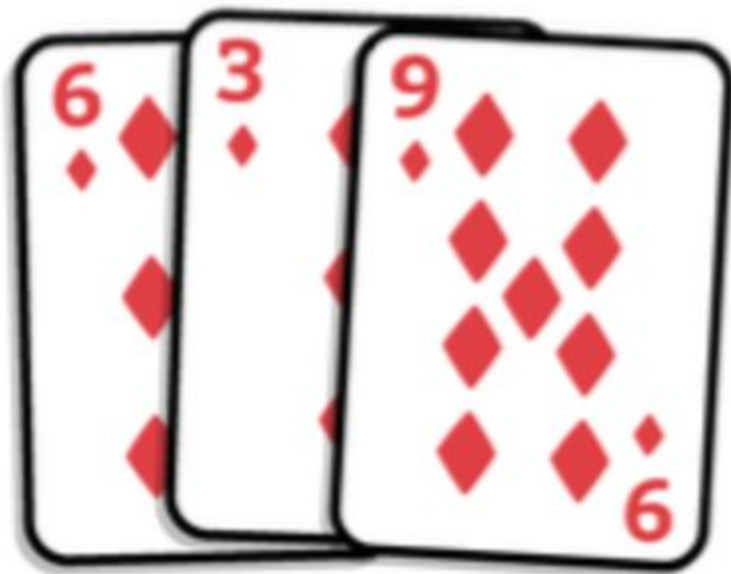
정렬되지 않은 카드 뭉치가 있다고 가정합니다.



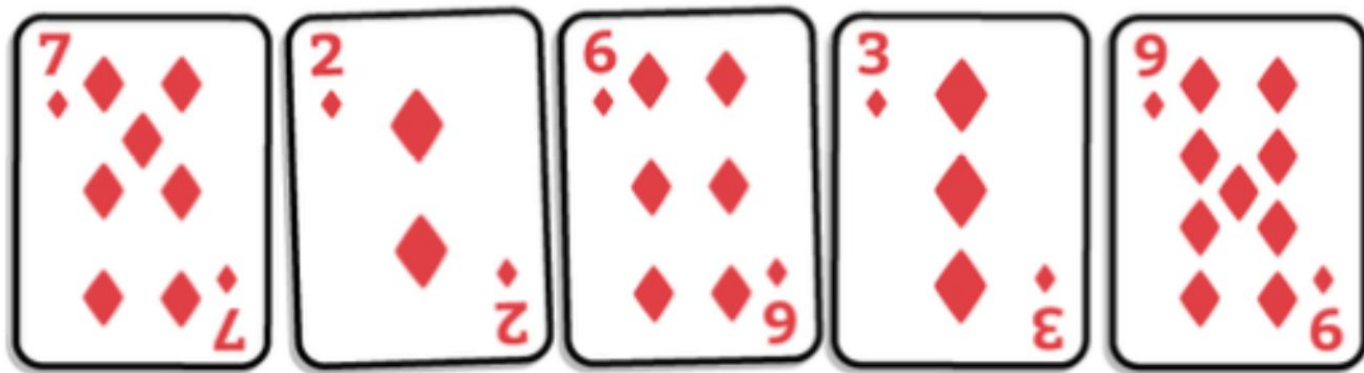
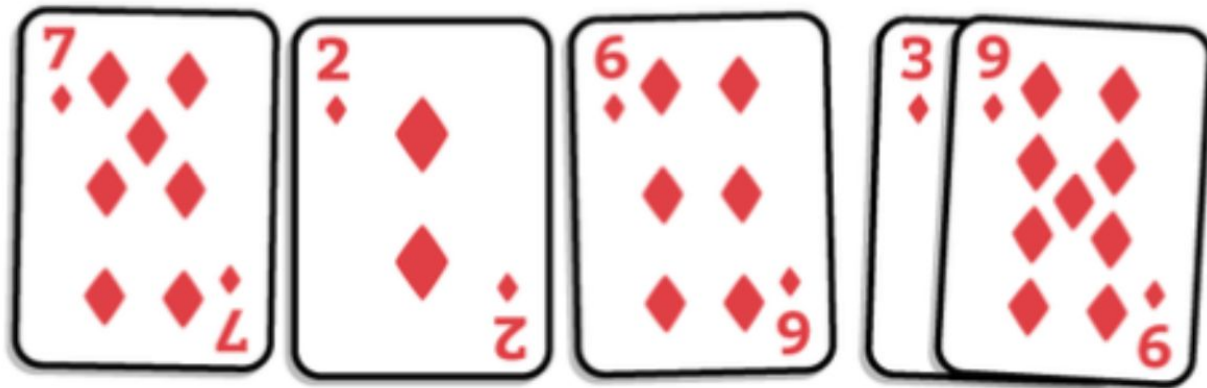
병합 정렬 알고리즘은 다음과 같이 작동합니다.



1. 먼저 반으로 나눕니다. 이제 정렬되지 않은 두 개의 더미가 있습니다.



2. 이제 더 이상 분리 할 수 없을 때까지 결과 더미를 계속 분할하십시오. 결국, 당신은 각 더미에 하나의 (정렬된) 카드를 갖게 될 것입니다



3. 마지막으로 파일을 분리한 역순으로 파일을 병합합니다. 병합 할 때마다 내용을 정렬 된 순서로 넣습니다. 각 개별 파일이 이미 정렬되어 있기 때문에 쉽습니다.



- 분할(Divide): 입력 배열을 같은 크기의 2개의 부분 배열로 분할한다.
- 정복(Conquer): 부분 배열을 정렬한다. 부분 배열의 크기가 충분히 작지 않으면 **순환 호출** 을 이용하여 다시 분할 정복 방법을 적용한다.
- 결합(Combine): 정렬된 부분 배열들을 하나의 배열에 합병한다.

초기상태 

21	10	12	20	25	13	15	22
----	----	----	----	----	----	----	----

Divide

21	10	12	20	25	13	15	22
----	----	----	----	----	----	----	----

Divide

21	10	12	20
----	----	----	----

25	13	15	22
----	----	----	----

Divide

21	10
----	----

12	20
----	----

25	13
----	----

15	22
----	----

Conquer  
Combine

21
----

10
----

12
----

20
----

25
----

13
----

15
----

22
----

Conquer  
Combine

10	21
----	----

12	20
----	----

13	25
----	----

15	22
----	----

Conquer  
Combine

10	12	20	21
----	----	----	----

13	15	22	25
----	----	----	----

← 정렬된 2개의 부분 리스트

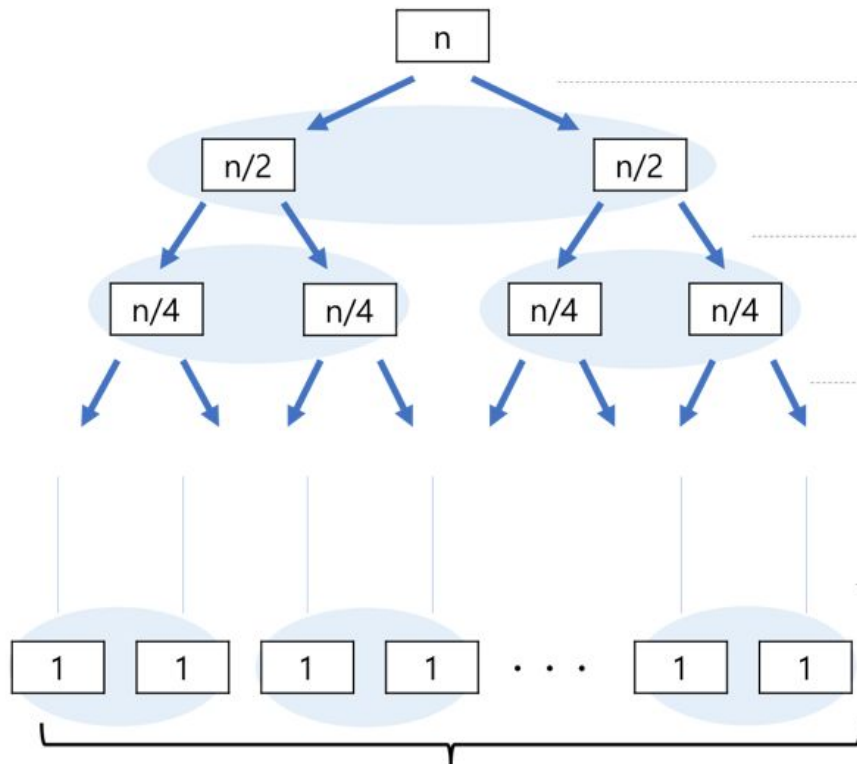
← 2개의 정렬된 리스트를 합병(merge)하는 단계  
(실제 정렬이 이루어지는 시점)

10	12	13	15	20	21	22	25
----	----	----	----	----	----	----	----

오름차순  
완성상태

10	12	13	15	20	21	22	25
----	----	----	----	----	----	----	----

순환 호출의 깊이  
(합병 단계의 수)  
 $k = \log_2 n$



크기  $n/2$ 인 부분 배열 1쌍 \*  $n$ 번 비교  
=  $n$ 번 비교

크기  $n/4$ 인 부분 배열 2쌍 \*  $(n/2)$ 번 비교  
=  $n$ 번 비교

⋮

크기 1인 부분 배열  $(n/2)$ 쌍 \* 2번 비교  
=  $n$ 번 비교

마지막 부분 배열의 개수:  $n$ 개