Introduction to Java - Algorithms

Recursion

Merge Sort Algorithm

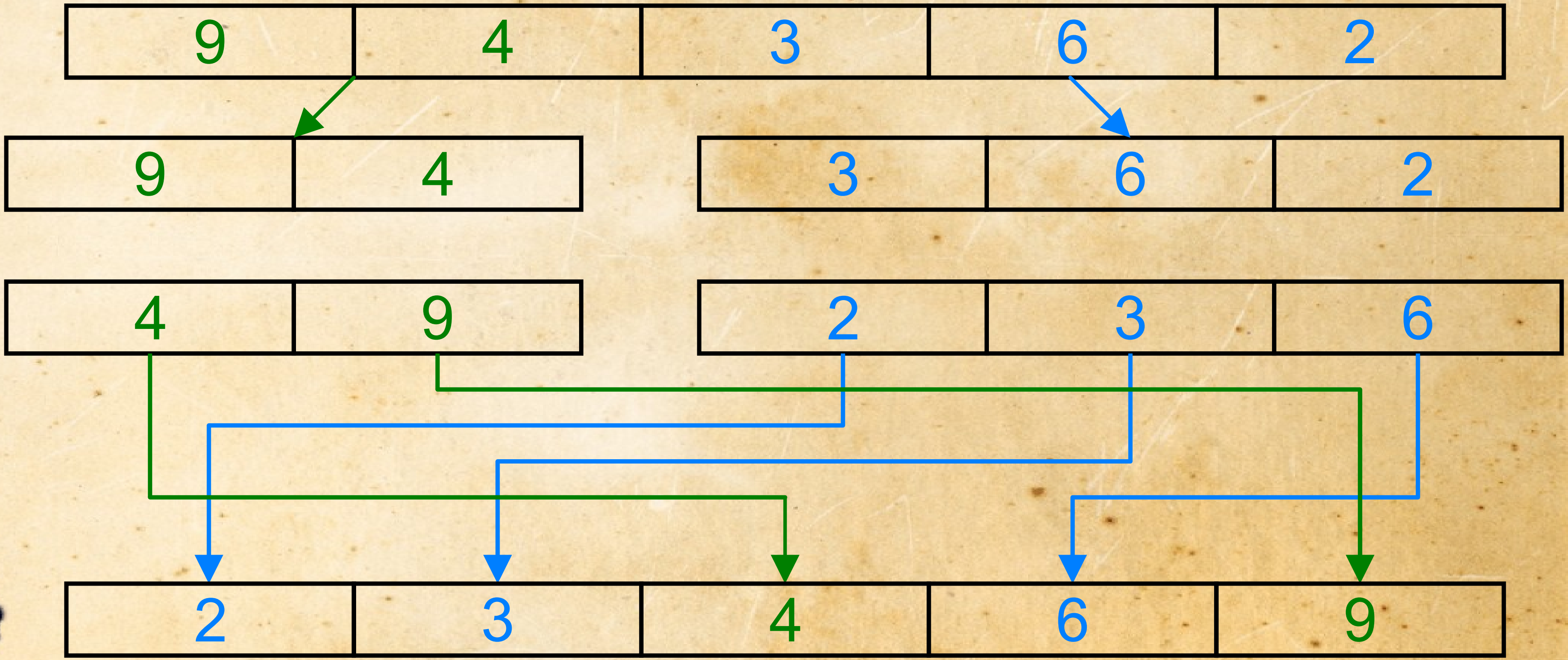
Lecture Contents

- Merge Sort
  - Uses
  - Algorithm

Merge Sort

- Pedagogical uses
  - Divide and conquer
  - Recursion
  - Algorithmic efficiency: $O(n \log n)$
    - Bubble sort is less efficient: $O(n^2)$
  - Sorting *stability*
- Real-World Uses
  - Traversing directories
  - Parsing languages

Merge Sort



9	4	3	6	2

9	4

3	6	2

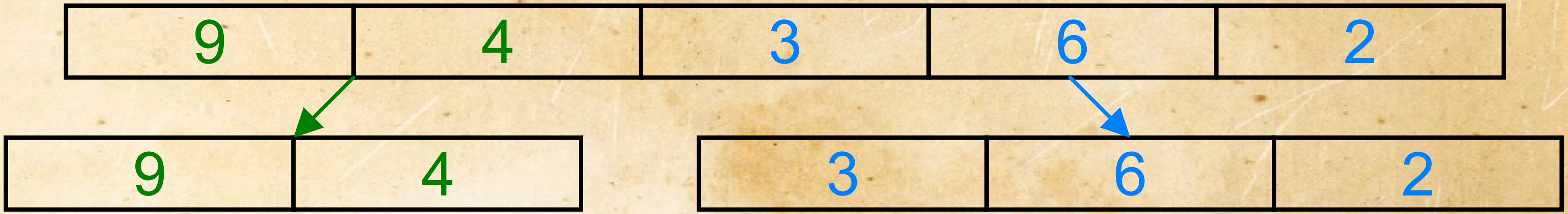
4	9

2	3	6

2	3	4	6	9

Merge Sort

- Divide and conquer…



9	4	3	6	2

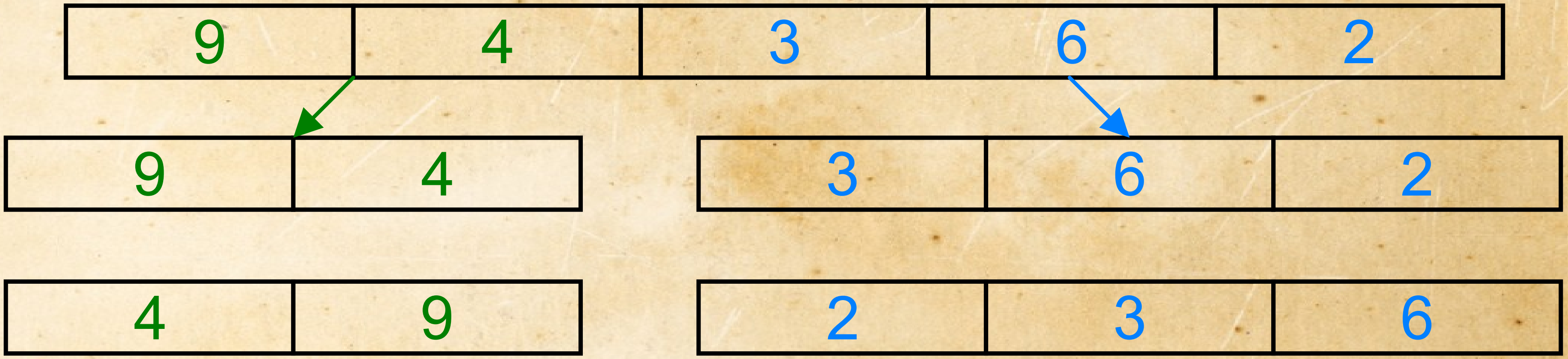
9	4

3	6	2

  - First we divide the array in half and call `mergeSort` on each half

Merge Sort

- Divide and conquer…



9	4	3	6	2

9	4

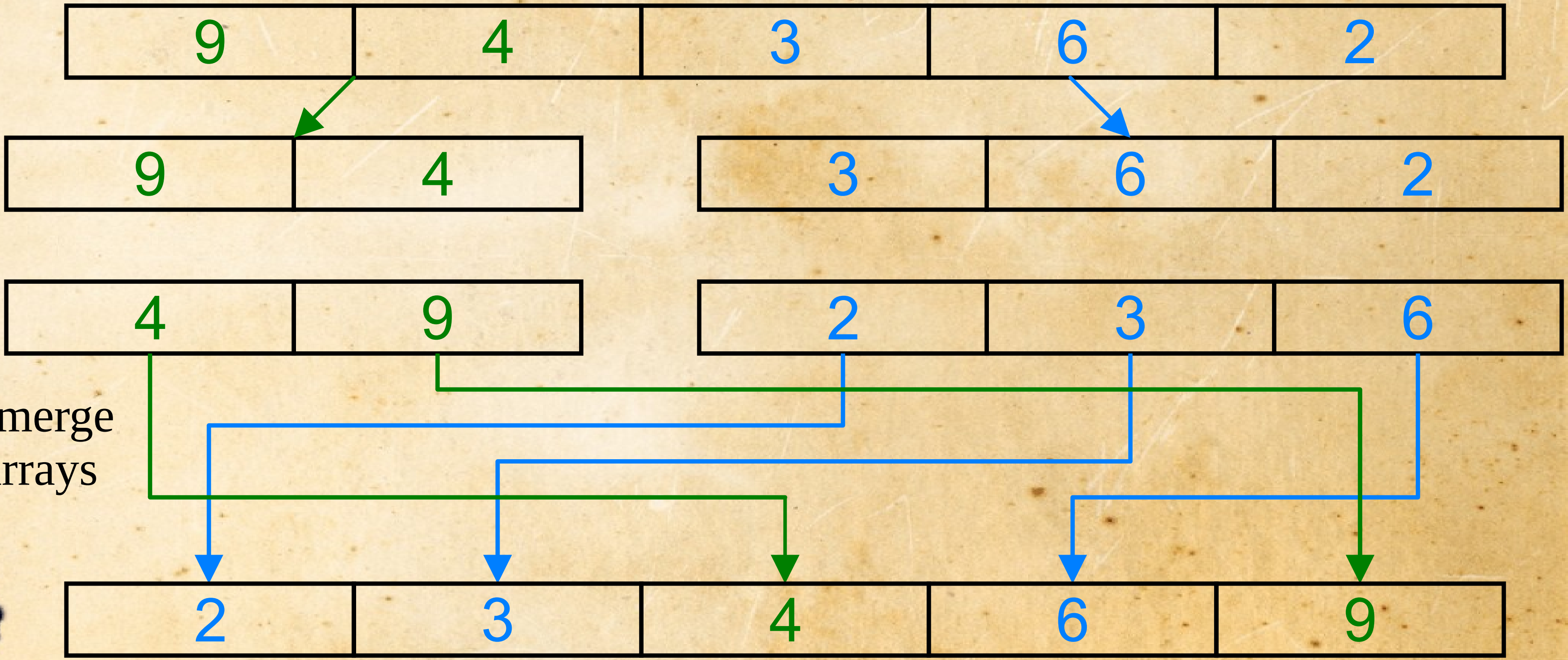
3	6	2

4	9

2	3	6

  - The `mergeSort` method returns sorted arrays (by recursion magic)

Merge Sort



9	4	3	6	2

9	4

3	6	2

4	9

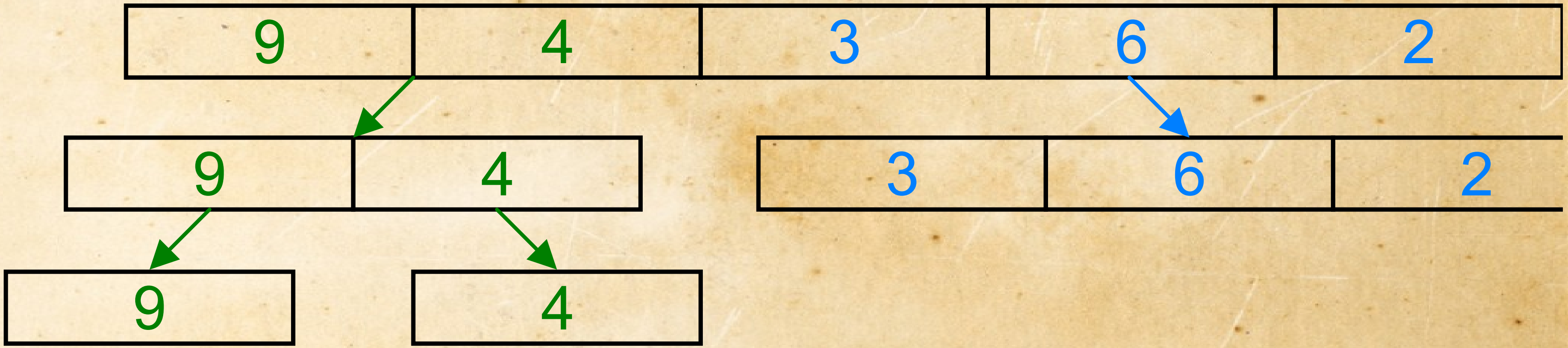
2	3	6

We then merge
the two arrays

2	3	4	6	9

Merge Sort

- The *terminating condition* is when the array has only one element.



9 4 3 6 2

9 4

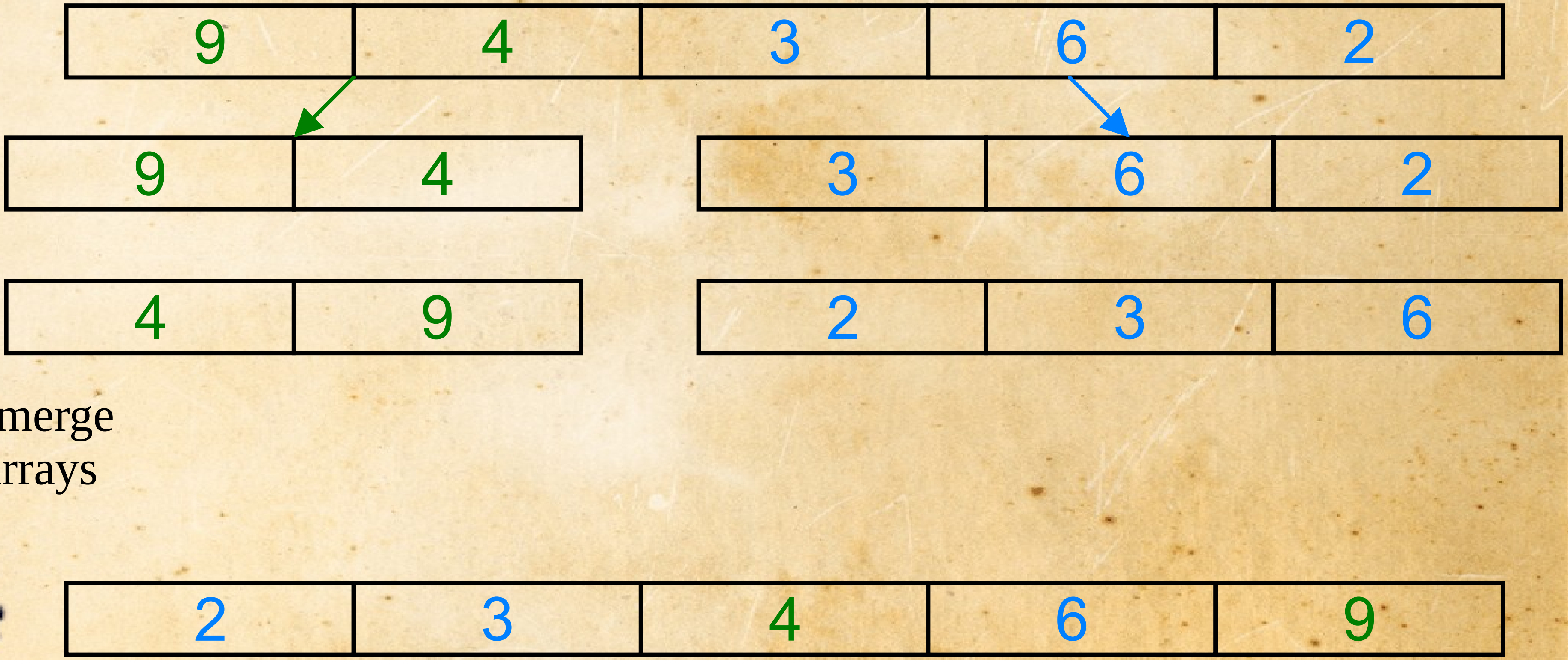
3 6 2

9

4

We then merge
the two arrays

Merge Sort



9 4 3 6 2

9 4

3 6 2

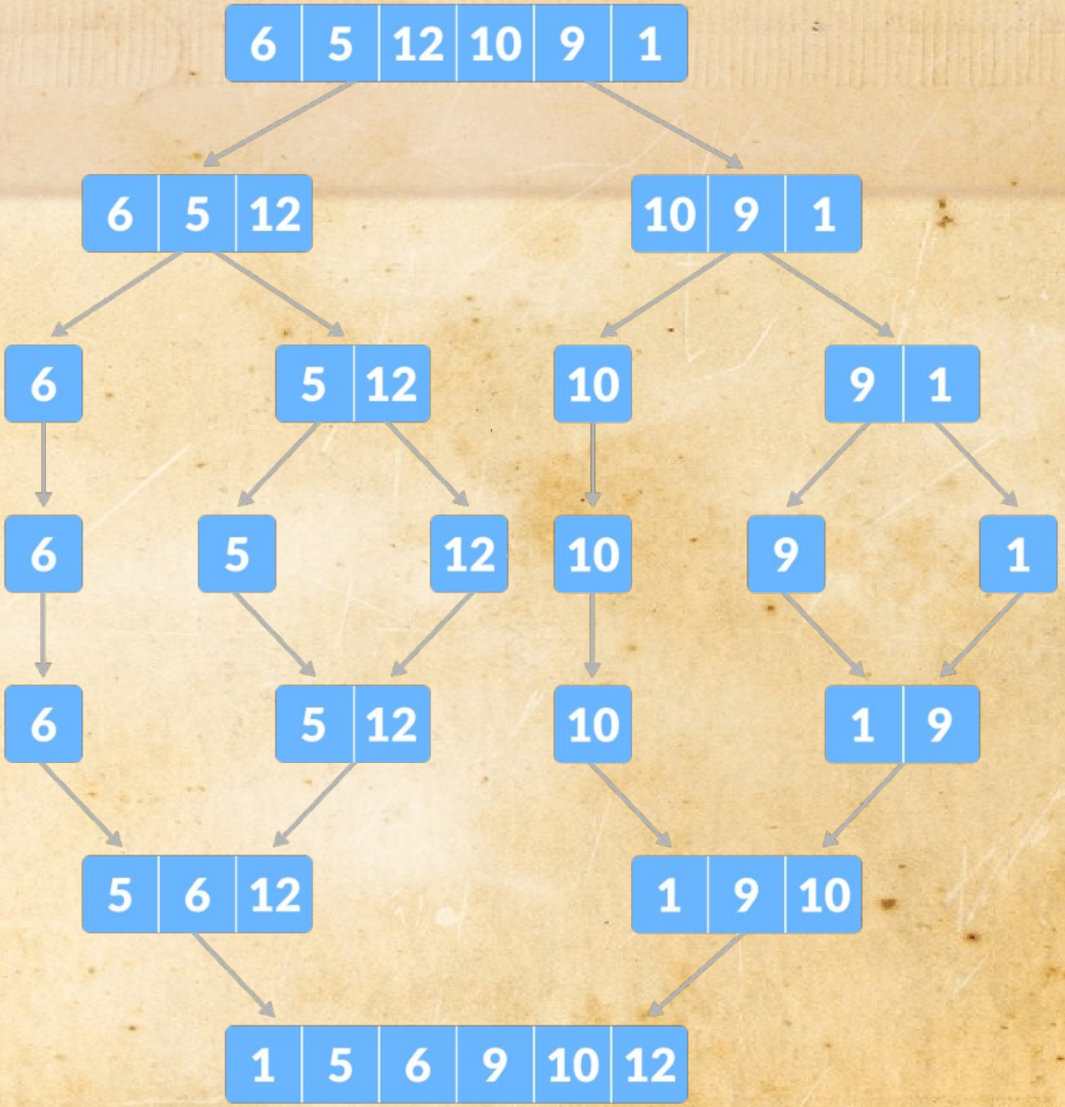
4 9

2 3 6

We then merge
the two arrays

2 3 4 6 9

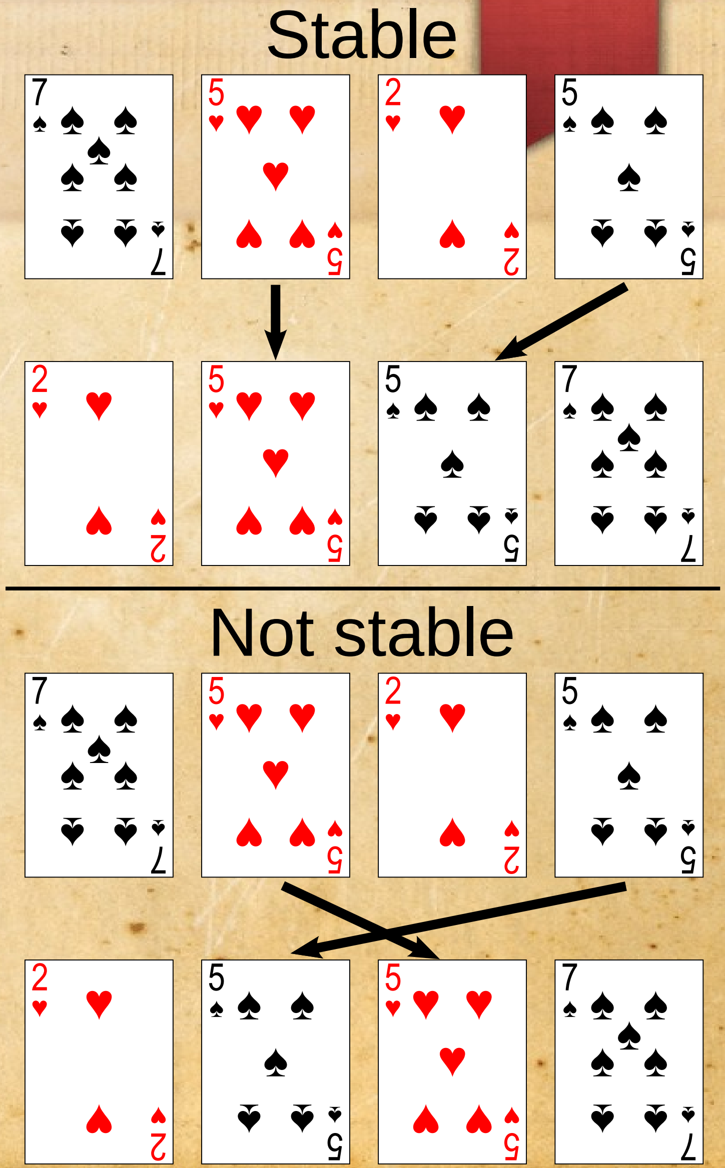
Merge Sort



6 5 12 10 9 1
6 5 12
10 9 1
6
5 12
10
9 1
6
5
12
10
9
1
6
5 12
10
1 9
5 6 12
1 9 10
1 5 6 9 10 12

Sorting Algorithm Stability

- A sorting algorithm is ***stable*** if it preserves the original order of elements that compare as equal
  - This is important if sorting will be done multiple times on the data set
    - For example sort cards by number, then sort them by suit. If the suit-sorting algorithm is stable, then the numerical order of the cards will be preserved.



Stable

Not stable

Recursion

Merge Sort Algorithm