

Recursion

Merge Sort Algorithm

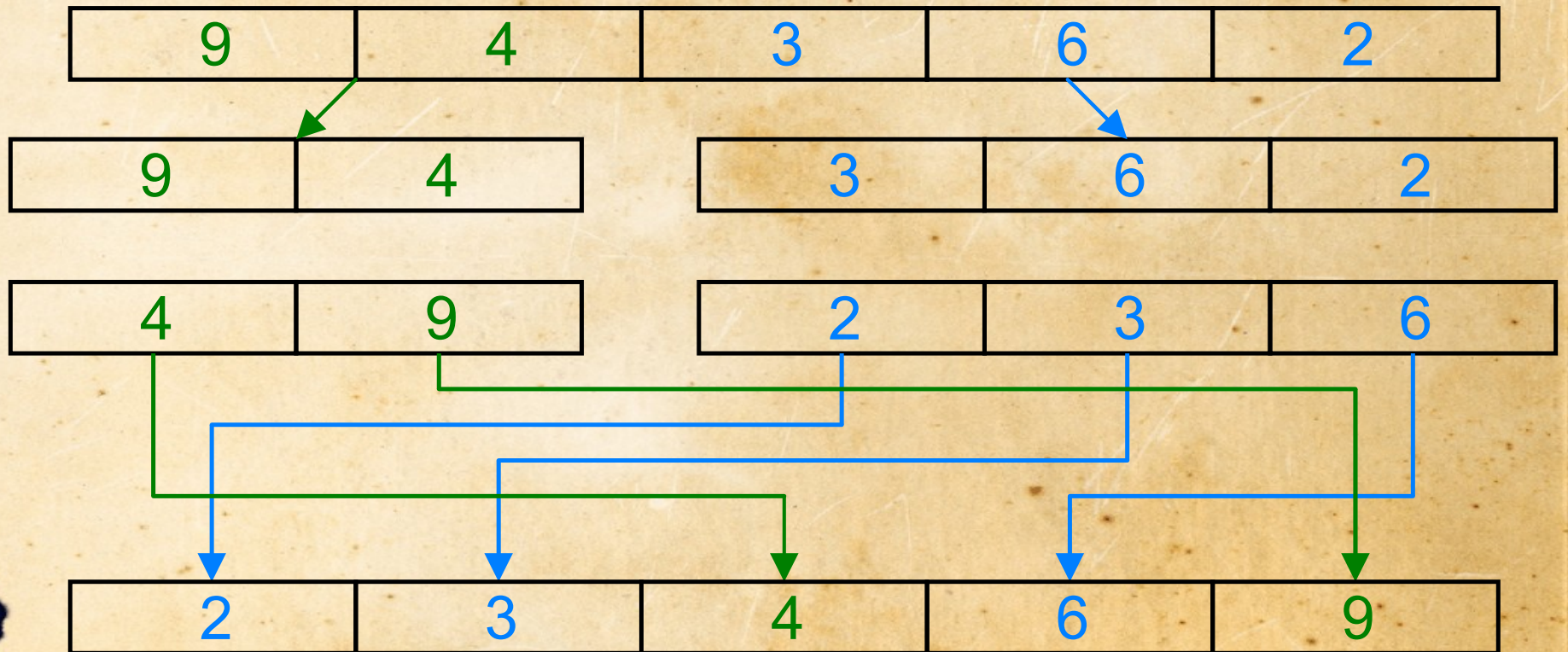
Lecture Contents

- Merge Sort
 - Pedagogical 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*

Merge Sort



Merge Sort

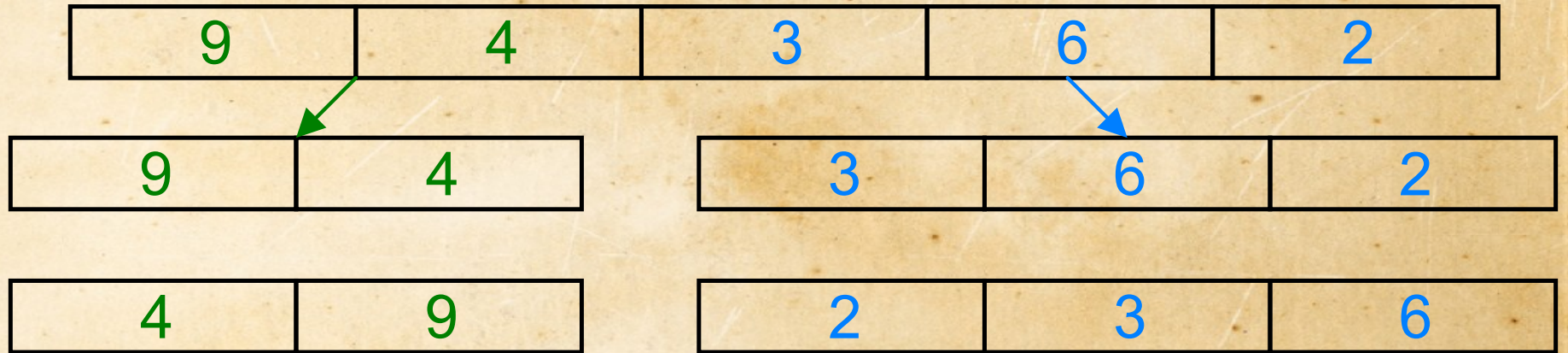
- Divide and conquer...



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

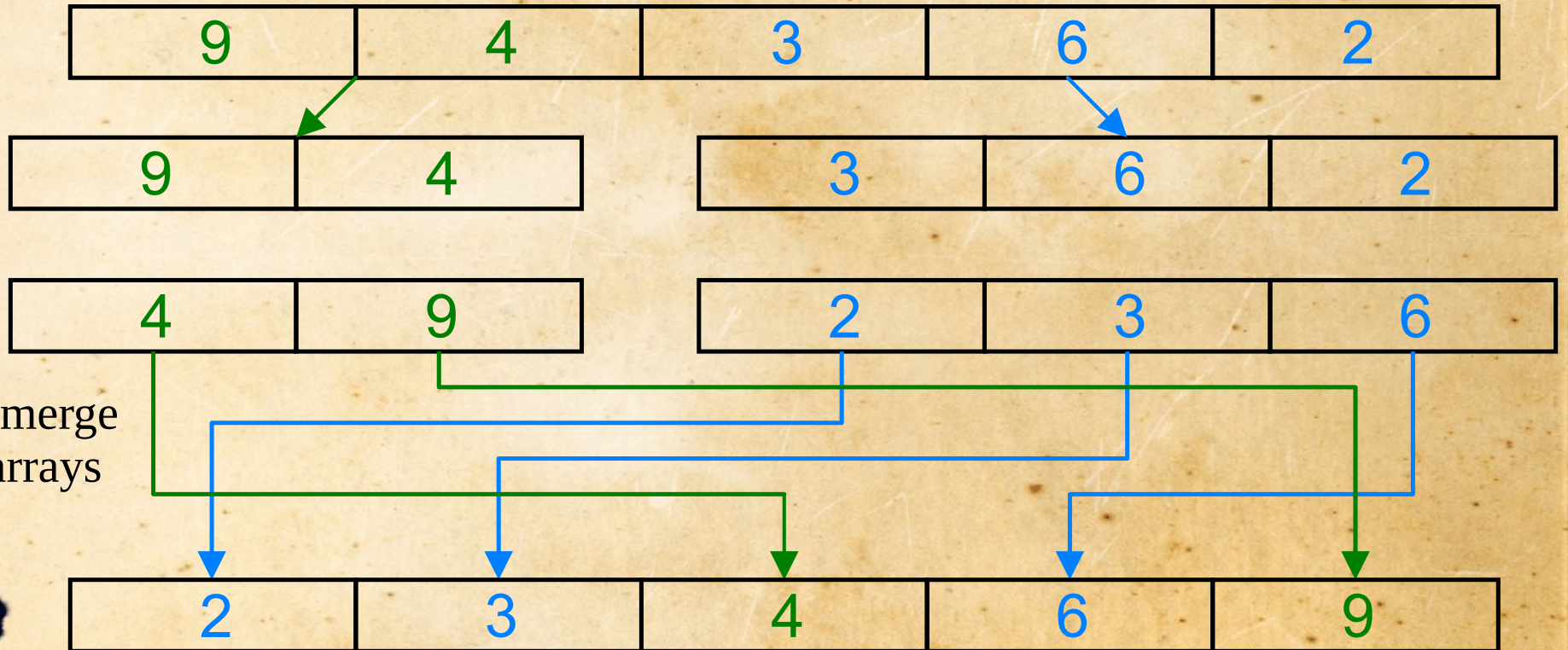
Merge Sort

- Divide and conquer...



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

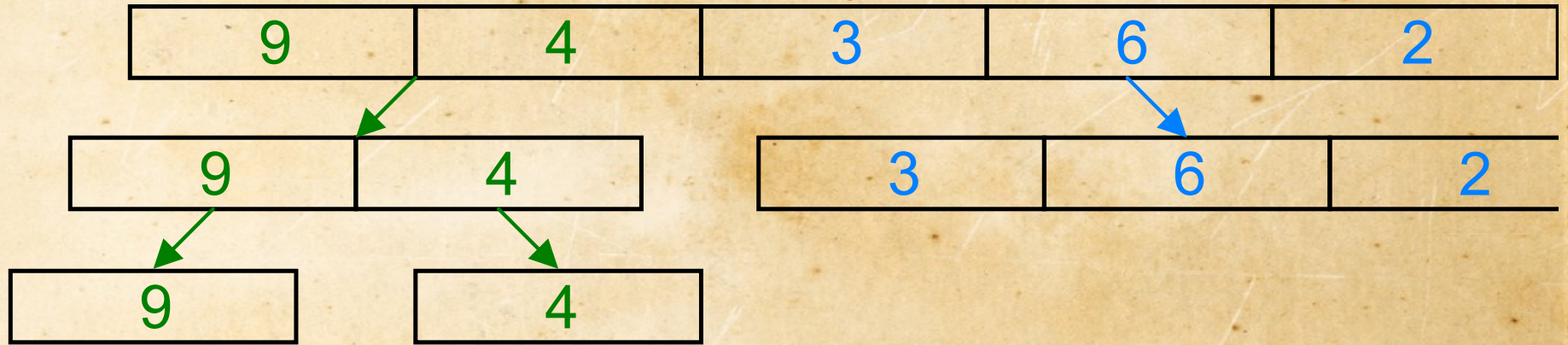
Merge Sort



We then merge
the two arrays

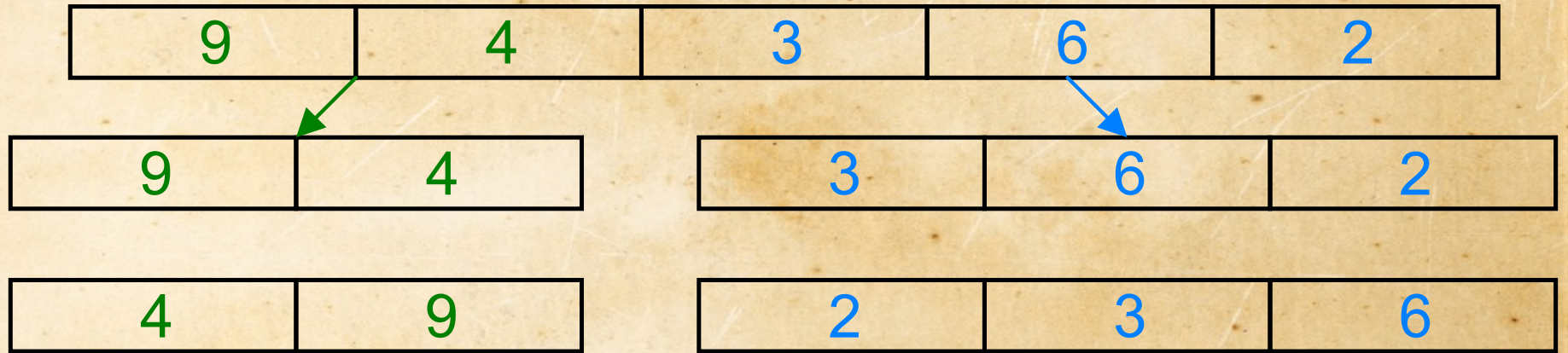
Merge Sort

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



We then merge
the two arrays

Merge Sort



We then merge
the two arrays

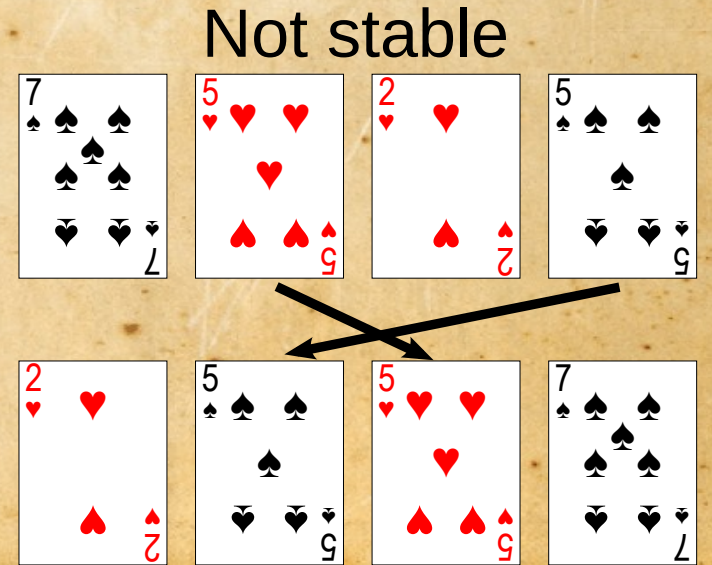
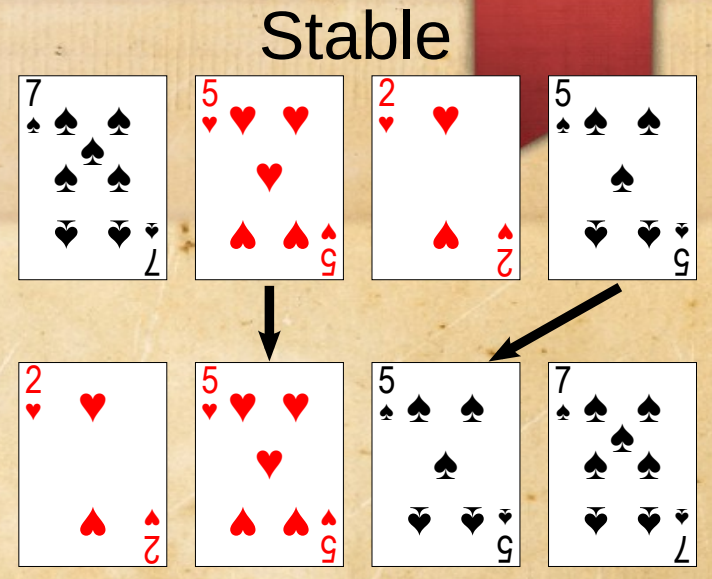


Merge Sort



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.



Recursion

Merge Sort Algorithm