# Merge Sort

Simulation & Time Complexity

# Merge Sort

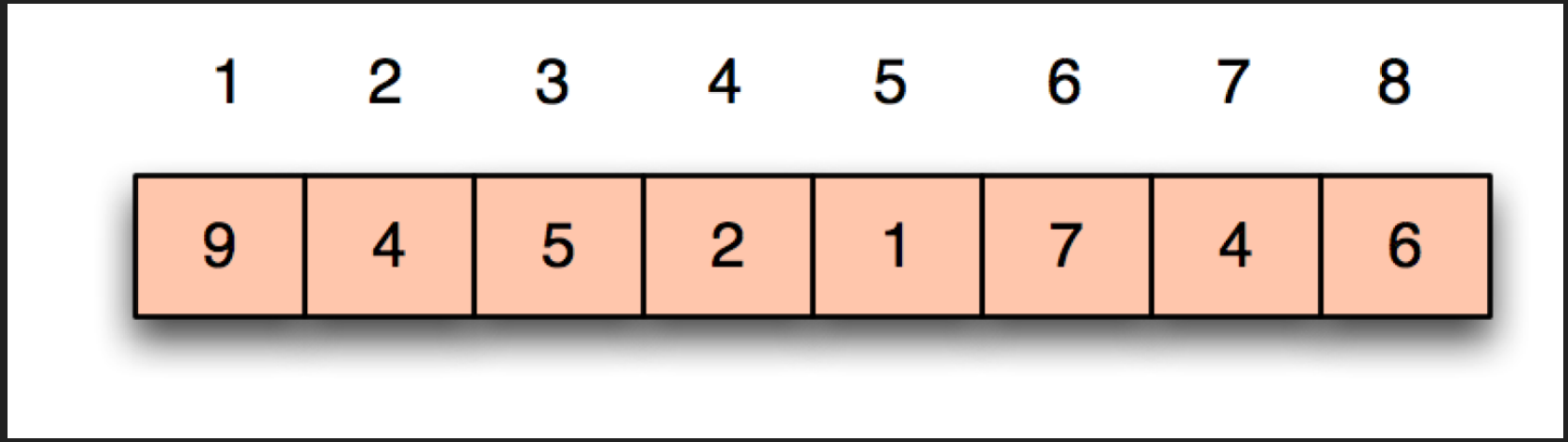
Sort an Array:

1. Divide: Trivial.
2. Conquer: Recursively sort 2 subarrays.
3. Combine: Merge the sorted subarrays in (n) time.

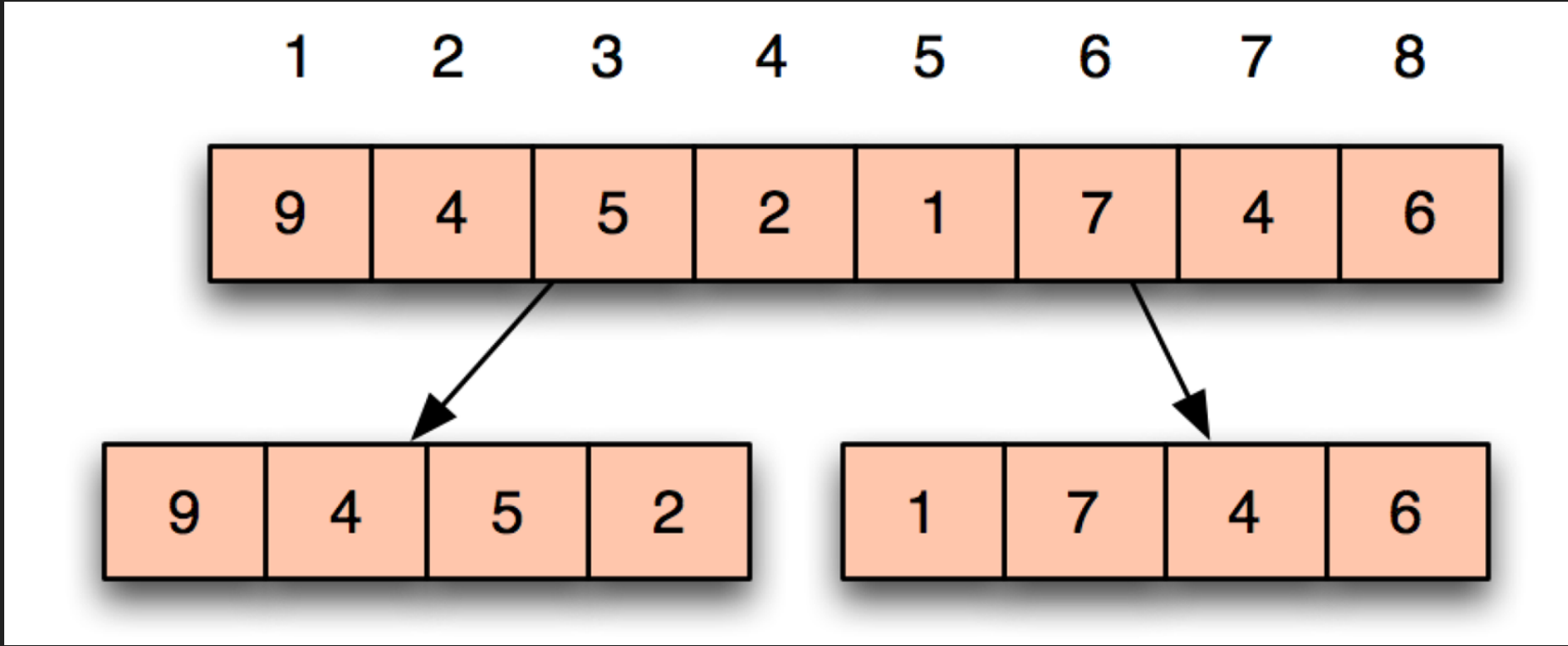
# Merge Sort Algorithm

```
MergeSort (Arr[], low, high) {

    mid = floor ((low+high)/2)

    if(low >= high){
        return Arr[low]
    }

    b = MergeSort(Arr[], low, mid)
    c = MergeSort(Arr[], mid+1, high)
    a = Merge (b,c)
    return a;
}
```

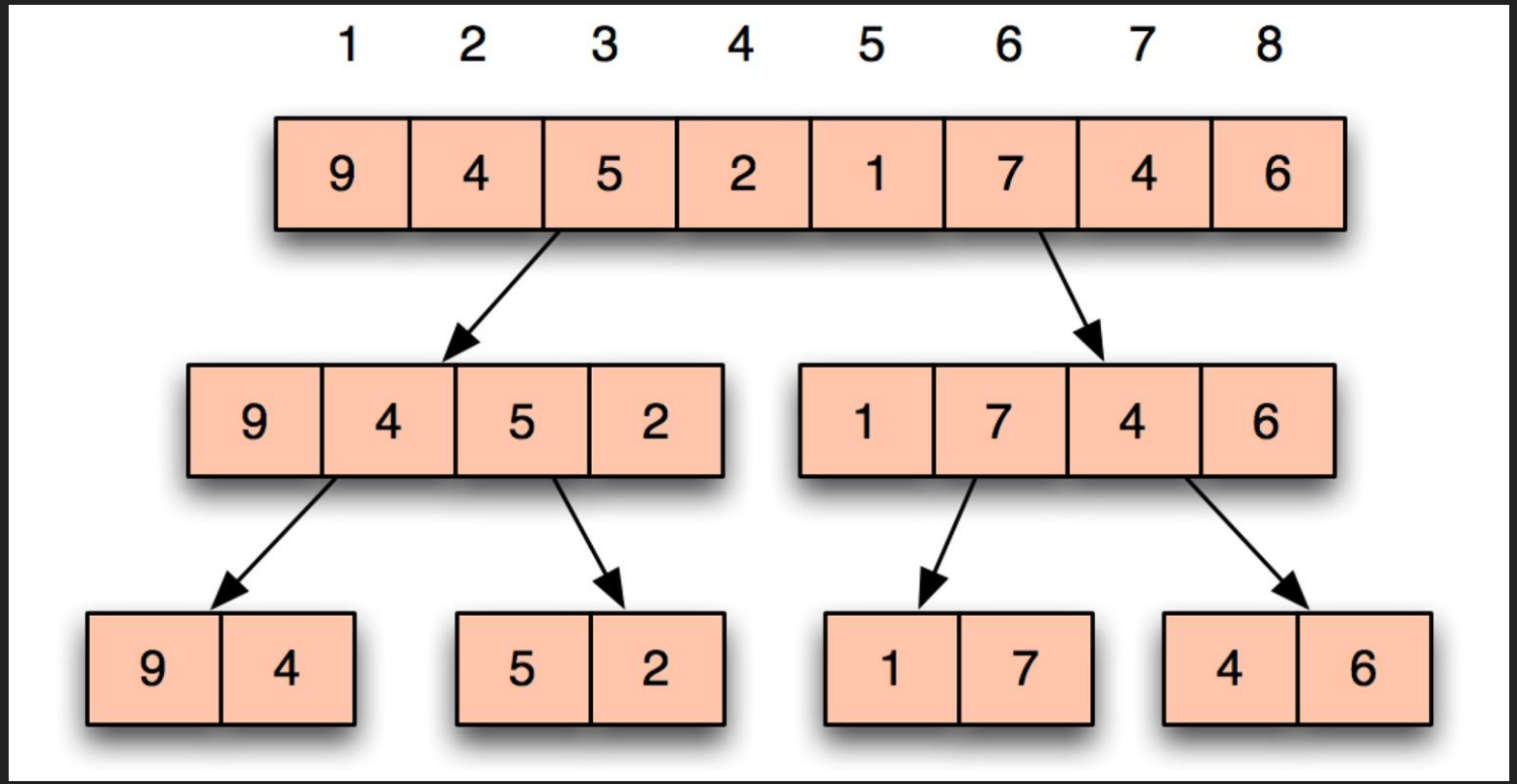
# Merge Sort In Action



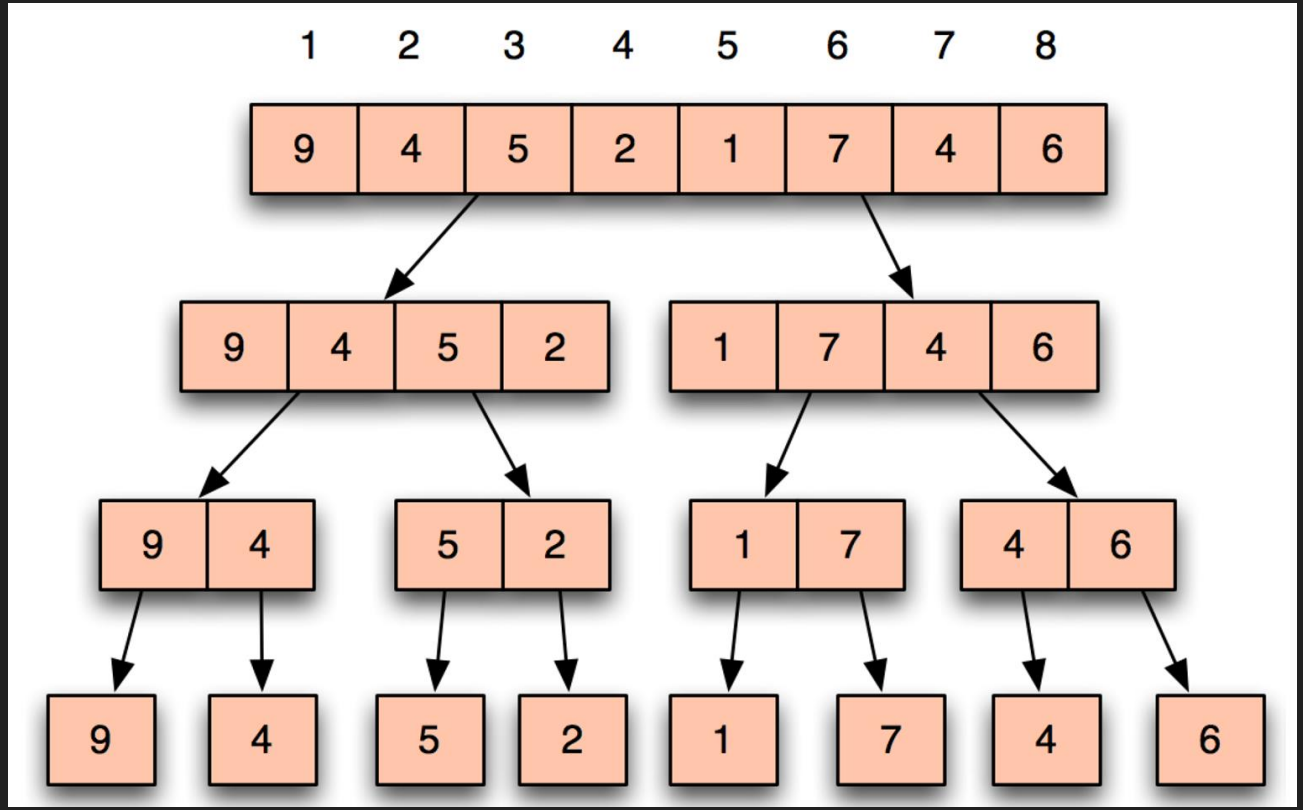
# Merge Sort In Action (II)



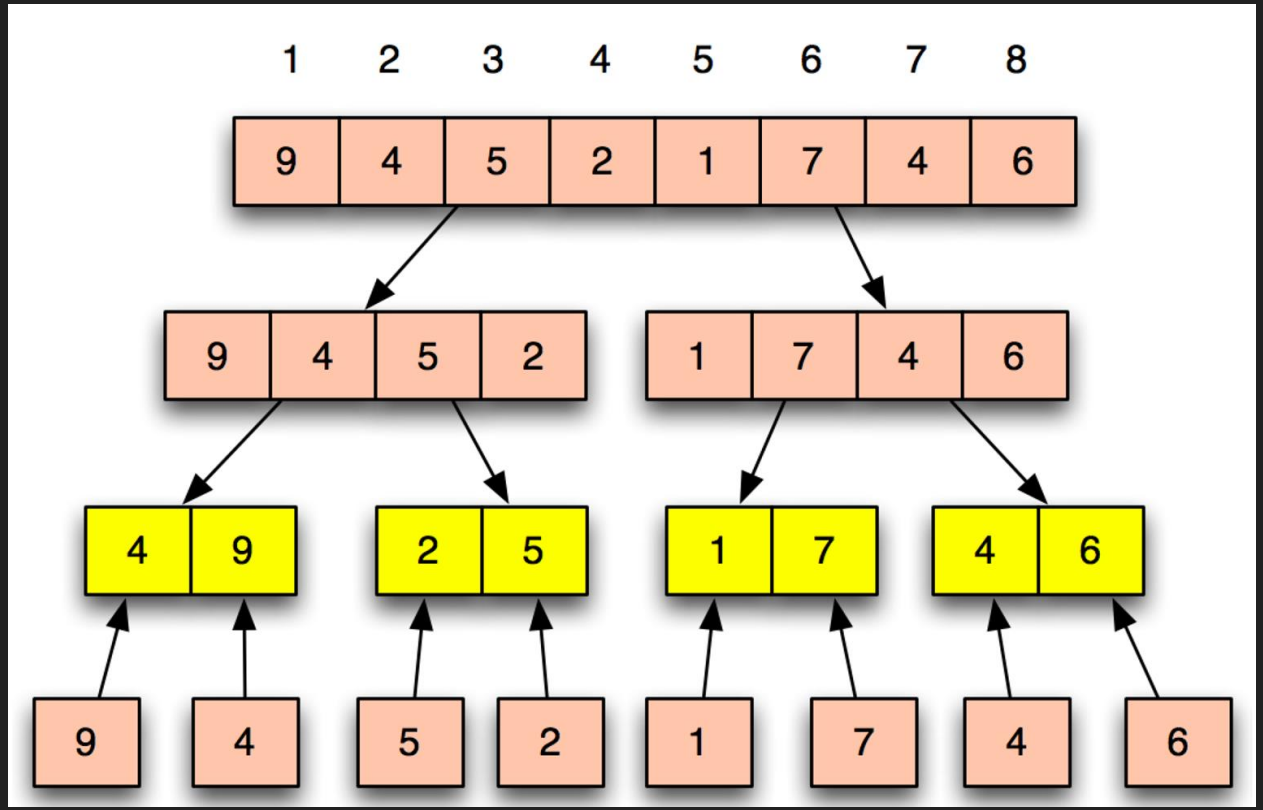
# Merge Sort In Action (III)



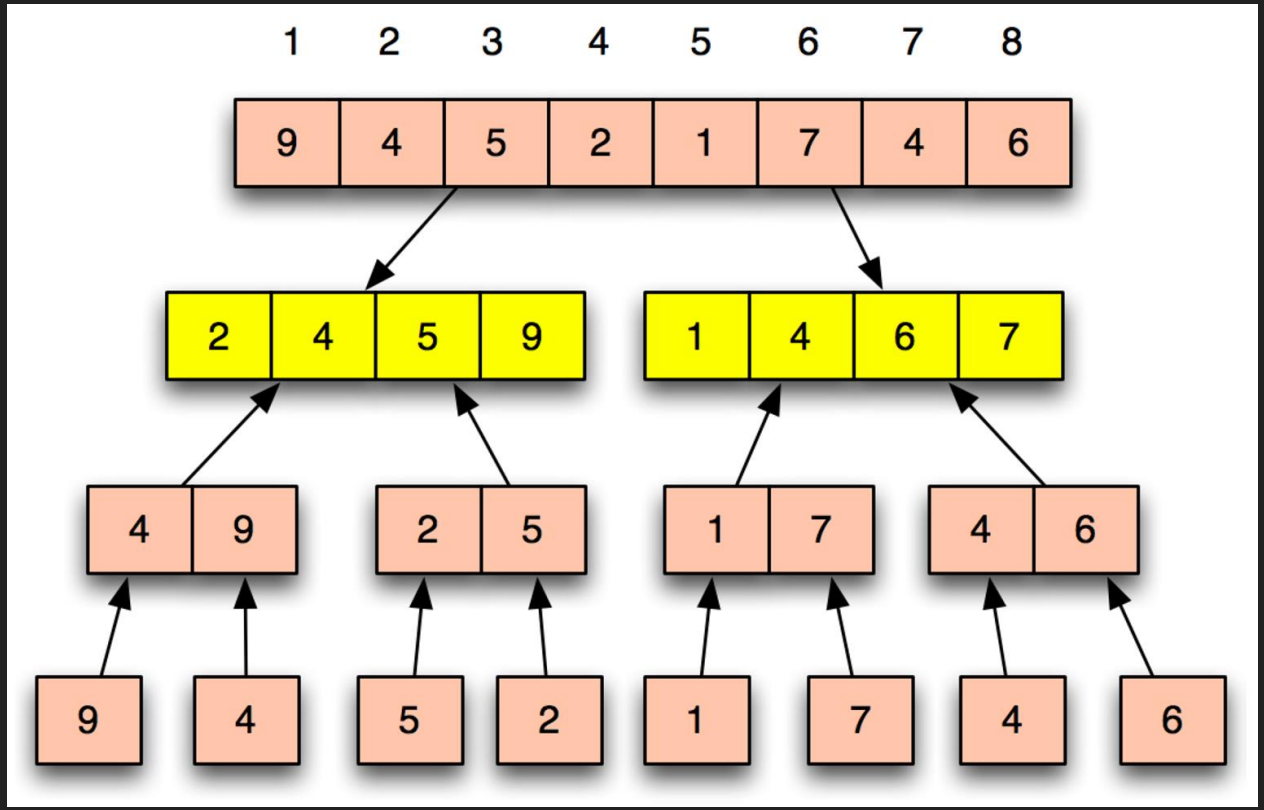
# Merge Sort In Action (IV)



# Merge Sort In Action (V)



# Merge Sort In Action (VI)



# Merge Sort In Action (VII)