QUICKSORT

CS340

Quicksort

- Like mergesort is a recursive algorithm
- **Divide:** Partition (rearrange) the array A[p..r] into two (possibly empty) subarrays A[p..q] and A[q+1..r] such that each element of A[p..q-1] is less than or equal to A[q], which is, in turn, less than or equal to each element of A[q+1..r]. Compute the index q as part of this partitioning procedure.
- **Conquer:** Sort the two subarrays A[p..q-1] and A[q+1..r] by recursive calls to quicksort.
- **Combine:** The subarrays are already sorted, no work is needed to combine them.

Quicksort

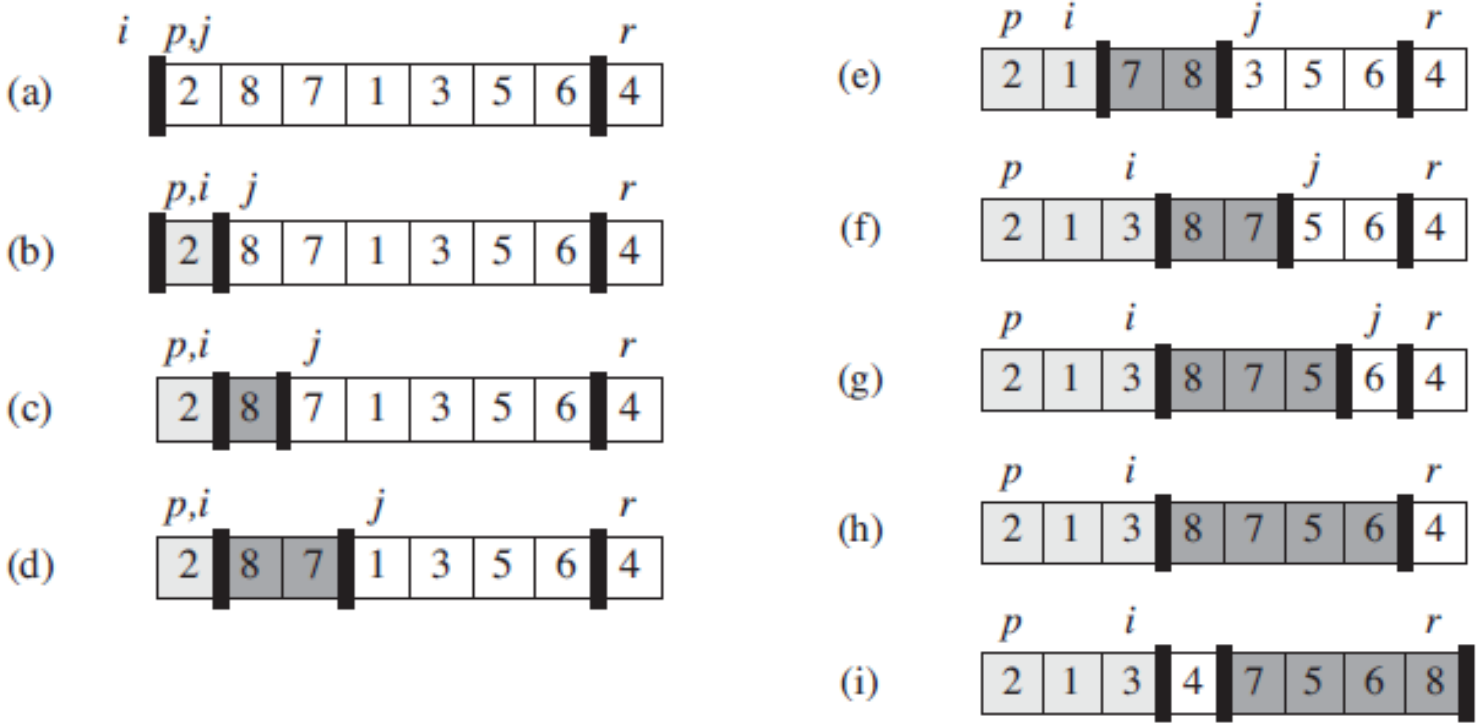
```
QUICKSORT(A, p, r)
1  if p < r
2      q = PARTITION(A, p, r)
3      QUICKSORT(A, p, q − 1)
4      QUICKSORT(A, q + 1, r)
```

To sort an entire array A, the initial call is QUICKSORT$(A, 1, A.length)$.

Partitioning the Array

```
PARTITION(A, p, r)
1  x = A[r]
2  i = p − 1
3  for j = p to r − 1
4       if A[j] ≤ x
5            i = i + 1
6            exchange A[i] with A[j]
7  exchange A[i + 1] with A[r]
8  return i + 1
```

Partitioning the Array



What is the time complexity of partitioning?

Quicksort time complexity

- What is the worst case?
- What is the best case?