CSE 201 1-30-20

Chapter 4: Divide & Conquer

Quicksort (actually in chap 7)

sort $A[p....r]$ recursively

Quicksort(A, p, r)

1. if $p < r$
2. $q = \text{Partition}(A, p, r)$
3. Quicksort$(A, p, q-1)$
4. Quicksort$(A, q+1, r)$

Partition causes :

$$A[p \ldots (q-1)] \le \underbrace{A[q]}_{\text{Pivot element}} < A[(q+1) \ldots r]$$

<u>Partition(A, p, q)</u>

```
i = p-1
for j = p to (r-1)
    if A[j] ≤ A[r]
        i++
        A[i] <-> A[j] (swap)
A[i+1] <-> A[r] (swap)
return (i+1)
```

What does Partition do?

$$\underbrace{A_p \cdots A_i}_{\le A_r} \; \underbrace{A_{i+1} \cdots A_{j-1}}_{> A_r} \; \underbrace{A_j \cdots A_{r-1}}_{\text{not yet compared}} \; \underbrace{A_r}_{\text{Pivot}}$$

- if $A_j \le A_r$: $A_j \leftrightarrow A_{i+1}$ (swap
  $i{+}{+}$ and $j{+}{+}$
- if $A_j > A_r$: $j{+}{+}$

Runtime: count # of array comparisons.

$T(n)$ = worst case # of comparisons on arrays of length n

At top level: Partition$(A, 1, n)$ does $n-1$ comparisons.

In worst case, $A[1 \ldots n]$ is already sorted.

$$T(n) = \begin{cases} 0 & n = 0, 1 \\ T(n-1) + (n-1) & n \geq 2 \end{cases}$$

Iteration

$$T(n) = (n-1) + T(n-1)$$
$$= (n-1) + (n-2) + T(n-2)$$
$$\vdots$$
$$= \sum_{i=1}^{k} (n-i) + T(n-k)$$

stop when $n-k=1$, so $k=n-1$

$$\therefore T(n) = \sum_{i=1}^{n-1} (n-i) + 0$$
$$= \sum_{i=1}^{n-1} n - \sum_{i=1}^{n-1} i$$
$$= n(n-1) - \frac{n(n-1)}{2}$$
$$= \frac{1}{2} n(n-1) = \Theta(n^2)$$