cse 102 10-26-21

Recurrence for MergeSort:

$$T(n) = \begin{cases} 0 & n = 1 \\ T(\lfloor \frac{n}{2} \rfloor) + T(\lceil \frac{n}{2} \rceil) + (n-1) & n \geq 2 \end{cases}$$

special case: n exact pow. of 2: 2^k

simplify to

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

Iteration method.

$$T(n) = (n-1) + 2T(\tfrac{n}{2})$$

$$= (n-1) + 2\left(\left(\tfrac{n}{2} - 1\right) + 2T\left(\tfrac{n/2}{2}\right)\right)$$

$$= (n-1) + (n-2) + 2^2 \cdot T(\tfrac{n}{2^2})$$

$$= (n-1) + (n-2) + 2^2\left(\left(\frac{n}{2^2}-1\right) + 2T\left(\frac{n/2^2}{2}\right)\right)$$

$$= (n-1) + (n-2) + (n-2^2) + 2^3 T\left(\frac{n}{2^3}\right)$$

$$\vdots$$

$$= \sum_{i=0}^{k-1} (n-2^i) + 2^k T\left(\frac{n}{2^k}\right)$$

$$= \sum_{i=0}^{k-1} n - \sum_{i=0}^{k-1} 2^i + 2^k T\left(\frac{n}{2^k}\right)$$

$$= kn - \frac{2^k - 1}{2-1} + 2^k T\left(\frac{n}{2^k}\right)$$

$$= kn - 2^k + 1 + 2^k T\left(\frac{n}{2^k}\right)$$

rec. terminates when: $\frac{n}{2^k} = 1$, i.e. $\boxed{k = \log_2(n)}$

for this k:

$$T(n) = n \cdot \lg(n) - 2^{\lg(n)} + 1 + 0$$

$$= n\lg(n) - n^{\lg(2)} + 1$$

$$\therefore \boxed{T(n) = n\lg(n) - n + 1}$$

exact solution when n = pow. of 2.

Exercise

write an algorithm (based on merge sort) that just checks to see if an array $A[p \ldots r]$ is sorted. (recursive)

- Prove its correctness by induction
- analyse runtime (ans: $\Theta(n)$)

note Iterative version:

A_1 A_2 A_3 A_4 $\cdots$ A_{n-1} A_n

compare, comp, comp, comp.

cmp $= n-1 = \Theta(n)$.

Exercise

Given $A = (A_1, A_2, \ldots, A_n)$, a pair of indices (i, j) is called an inversion if: $i < j$ and $A_i > A_j$.

write an algorithm (based on merge sort) that counts the # of inversions.

note: # of pairs of distinct indices $= \binom{n}{2} = \frac{n(n-1)}{2}$

- write algorithm
- Prove correctness (induction)
- analyze runtime

note: sort as you go: $\Theta(n \log n)$ ← ** do this!

don't sort: $\Theta(n^2)$

Quicksort (sec. 7.1 - 7.4)

again $A[p \ldots r]$ is to be sorted.

Quicksort(A)

	cost when $r=n, p=1$
1. if $p < r$	0
2. $q = \text{Partition}(A, p, r)$	$n-1$
3. Quicksort $(A, p, q-1)$	$T(n-1)$
4. Quicksort $(A, q+1, r)$	$T(0)$

Partition (A, p, r) re-arranges $A[p \ldots r]$ and returns index q s.t.

$$\underbrace{A[p \ldots (q-1)]}_{\text{not sorted}} \leq \underbrace{A[q]}_{\text{Pivot element}} < \underbrace{A[(q+1) \ldots r]}_{\text{not sorted}}$$

q : Pivot index

Pick Pivot to be rightmost: $A[r]$

Partition (A, p, r)

1. $i = p-1$
2. for $j = p$ to $(r-1)$
3. if $A[j] \leq A[r]$
4. $i += 1$
5. $A[i] \leftrightarrow A[j]$ # swap
6. $A[i+1] \leftrightarrow A[r]$ # swap
7. return $(i+1)$

Invariants:

$\underbrace{A_p \cdots A_i}_{\leq \text{Pivot}}$ $\underbrace{A_{i+1} \cdots A_{j-1}}_{> \text{Pivot}}$ $\underbrace{A_j \cdots A_{r-1}}_{\text{unknown}}$ $\underbrace{A_r}_{\text{Pivot}}$

If $A_j \leq A_r$: swap $A_j \leftrightarrow A_{i+1}$, $i{+}{+}$, $j{+}{+}$

If $A_j > A_r$: $j{+}{+}$

Runtime of Pivot:

same in best, worst & avg cases

$\#$ comparisons $= (r-p+1)-1 = r-p$

at top level: $\#$ comp $= n-1$

Exercise

Prove correctness of quicksort. Assume correctness of Partition()

Runtime of Quicksort:

- worst case: $\Theta(n^2)$
- avg. case: $\Theta(n \log n)$

worst case

occurs when array $A[1 \ldots n]$ is already sorted. Partition$(A, 1, n)$ gives:

$$A[1 \text{-----} (n-1)] \leq \underbrace{A[n]}_{\text{Pivot}} < \ldots \text{empty} \ldots$$

$$q = n$$

let $T(n)$ denote worst case # comp. performed by quicksort on $A[1 \ldots n]$.

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

By iteration method

$$T(n) = (n-1) + T(n-1)$$

$$= (n-1) + (n-2) + T(n-2)$$

$$= (n-1) + (n-2) + (n-3) + T(n-3)$$

$$\vdots$$

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

halt when $n-k = 1$, i.e. $\boxed{k = n-1}$

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

$$= n(n-1) - \frac{n(n-1)}{2}$$

$$\boxed{T(n) = \frac{1}{2} n(n-1)} \leftarrow \text{exact soln}$$

$$\therefore \boxed{T(n) = \Theta(n^2)} \leftarrow \text{asymp. soln}$$

Average Case

Assume all $n!$ permutations of $A[1 \ldots n]$ are equally likely, i.e uniform prob. distribution.

let $t(n) = (\text{avg. \# comp.})$ performed by quicksort on $A[1 \ldots n]$. i.e.

$$t(n) = \frac{\sum_{\text{all permutations}} \left(\begin{array}{l} \text{\# of comp. by Q.S.} \\ \text{on given permutation} \end{array} \right)}{n!}$$

Goal: write a recurrence relation for $t(n)$. will solve exactly, then extract asym. solution.