CSE 101 – Oct 2, 2019 (Week 1)

Notes provided by Ben Sihota bsihota@ucsc.edu

Review

- Justified dropping lower order terms
- Want to release ourselves from the dependence on coefficients
- Some algorithms are "free" – uses other tricks and has other limitations
- Number of array comparisons is the "currency"

How to Analyze Recursive Algorithms

Problem instance, I

Split into sub-instances *(divide stage)*

$I_1, I_2, ..., I_k$

Sub-instance solutions, $S_1, S_2, ..., S_k$

~ *Make the problem so small that solving it is trivial*

Assemble the sub-instance solutions into the **problem solution**, S *(conquer stage)*

Merge Sort

$$A[p...r]$$

~ *0 and 1 length arrays are already sorted; if $p = r$, then length = 1*

(unsorted)

$$A[p...q] \quad A[q+1...r]$$

$$\downarrow \quad \downarrow$$

(sorted)

$$A[p...q] \quad A[q+1...r]$$

$$\downarrow \quad \downarrow \text{merge}$$

(copy)

$$A[1...(q-p+1)] \quad A[1...(r-q)]$$

$$\downarrow$$

$$A[p...r]$$

(sorted)

$q =$ index in middle $= [\frac{p+r}{2}]$

Merge Sort Pseudocode

Mergesort(A, P, R):

1. if $p < r$
   2. $q = [\frac{p+n}{2}]$
   3. MergeSort(A, p, q)
   4. MergeSort$(A, q+1, r)$
   5. Merge(A, p, q, r)

Let $T(n) = \#$ of array comparisons performed by merge sort on arrays of length n *(worst case)*

Cost:

Step (see above)	Cost
1	0
2	0
3	$T([\frac{n}{2}])$
4	$T([\frac{n}{2}])$
5	n-1

Notice: The top level call:

MergeSort$(A, 1, n)$

i.e. $p = 1$, $r = n$

So $q = [\frac{1+n}{2}]$

-> $\therefore$ *length*$(A[p...q]) = \frac{n}{2}$

Exercise: Show ↑ and ↓

and

-> length$(A[q+1...r]) = \frac{n}{2}$

~ *Merge sort pseudocode is in the textbook*

Thus

$$T(n) = \begin{vmatrix} 0 & n = 1 \\ T([\frac{n}{2}]) + T([\frac{n}{2}]) + (n-1) & n \geq 2 \end{vmatrix}$$

To **solve**, find closed formula

$T(n) = \ldots$ some exp. in n only (exact solution)

Asymptotic solution: $T(n) = \Theta(...)$

Asymptotic Solution

Trick: Restrict $n =$ exact power of 2, i.e.

$$n = 2^k$$

$\therefore \lfloor \frac{n}{2} \rfloor = \lceil \frac{n}{2} \rceil = \frac{n}{2}$

Recurrence becomes:

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

(n is a power of 2)

Exact solution:

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

Check: Rhs $= 2T(\frac{n}{2}) + (n-1)$

$= 2[\frac{n}{2} * lg(\frac{n}{2}) - \frac{n}{2} + 1] + (n-1)$

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

$= n(lgn - lg2) + 1$

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

Asymptotic Solution $T(n) = \Theta(nlgn)$

Fact: This is the asymptotic solution even when n is not a power of 2