Karatsuba, Quickselect

Thursday, February 27, 2020 3:30 PM

H(n) LLL

H(n-1) H(n-1)

H(n-2) H(n-2) ...

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

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

M(n)

$M(n/2) + M(n/2) + O(n)$

O(n) n

$T(n) = 2T(n/2) + O(n)$

$T(n) = O(n \log n)$

Today:

Divide and Conquer "self reduction but effective"

Master's Theorem

Thm: $T(n) = aT(n/b) + f(n)$

- If $a \cdot f(n/b) = K f(n)$ for $K<1$

  $T(n) = \Theta(f(n))$

  "most of the work at the root of the tree"

- If $a \cdot f(n/b) = K f(n)$ for $K>1$

  $T(n) = \Theta(n^{\log_b a})$

  "most of the work done in the leaves"

- If $a \cdot f(n/b) = f(n)$

  $T(n) = \Theta(f(n) \cdot \log n)$

  "same work in every level"

e.g. merge sort

$T(n) = 2T(n/2) + n$

$= a \cdot T(n/b) + f(n)$

for $a=2$, $b=2$, $f(n)=n$

$2 \cdot f(n/2) = 2 \cdot n/2 = n = f(n)$

High school mult: A primitive operation is × of single digit.

3457
567

n digit numbers.

Break 3457 into $3000 + 400 + 50 + 7$

product $\underbrace{3000 \cdot 567}_{O(n)} + \underbrace{400 \cdot 567}_{O(n)} + 50 \cdot 567 + 7 \cdot 567$

mult(x, y): $|x| = |y| = n$ digits

acc := 0

for each i'th digit of x: O(n)

acc += $10^i (X[i] \times y)$ O(n)

$T(n) = O(n^2)$

$3457 = (34 \cdot 10^2 + 57)$

$\times\ 0567 \quad \times (5 \cdot 10^2 + 67)$

$(a \cdot 10^i + b)(c \cdot 10^i + d)$ foil

$ac \times 10^{2i} + (ad + bc) \cdot 10^i + bd$

mult2(x, y, n): // x, y are n digit #

if n = 1 then return x·y (1 operation)

else: a, b := X[1...n/2], X[n/2...n]

c, d := y[1...n/2], y[n/2...n]

let, ac = mult2(a, c, n/2)

$$4\times \begin{cases} ad = \text{mult2}(a, d, n/2) \\ bc = \text{mult2}(b, c, n/2) \\ bd = \text{mult2}(b, d, n/2) \end{cases}$$

return $a0 \cdot 10^i$ ↑ see above

$T(1) = 1$

$T(n) = 4 \cdot T(n/2) + n$

$= a \cdot T(n/b) + f(n)$

use $a=4$ $b=2$ $f(n)=n$

$a \cdot f(n/b) = 4\,n/2 = 2 f(n)$

$= \Theta(n^{\log_2 4}) = \Theta(n^2)$

$(a \cdot 10^i + b)(c \cdot 10^i + d)$

$ac \cdot 10^{2i} + (bc + ad) \cdot 10^i + bd$

$bc + ad = \cancel{ac} + \cancel{bd} - (a-b)(c-d)$

$\cancel{-ac} \cancel{-bd} + bc + ad$

$T(n) = 3 \cdot T(n/2) + n$

$\Theta(n^{\log_2 3}) = \Theta(n^{1.58})$

mult3(x, y): $ac :=$ ...
$bd :=$ ...
⋮
$bc =$ ⋮
$ad =$
$m = (bc + ad) := ac + bd - \text{mult3}((a-b), (c-d), n/2)$

Select (k, arr): select the k'th smallest element in an unsorted array.

select (1, arr) ?
"linear time"

// ||arr|| = n

Select (3, arr)

ind ← Select(1, arr)
arr[ind] := ∞
$3 \cdot T(1, n) = 3n$

Select (n/2, arr)

$\therefore O(n^2)$

arr' := Sort(arr) // merge sort
// $O(n \log n)$

arr'[k]

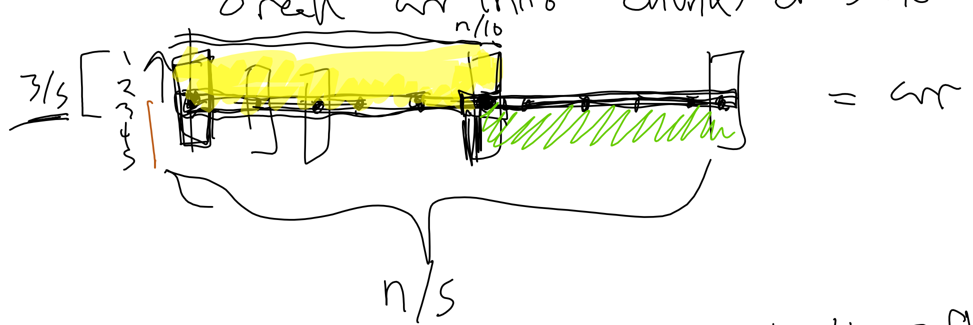
mom select(k, arr, |Arr| = n):

median(arr) ~ select(n/2, arr)

if n is small enough: O(1)
do it directly

otherwise:

break arr into chunks of 5 elements



- find median of each chunk of five
each of these takes O(1)
and n/5 of them. O(n) total.

r := mom select(n/2, all median of fives, n/5)
row[2]

T(n/5)

yellow region guaranteed smaller than r.

At least 3n/10 of the values (the yellow region) are smaller than r.

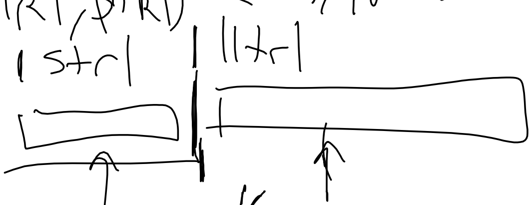
- partition(arr, r):

smallerThanR, largerThanR := part(arr, r)

|STR| + |LTR| = n-1

min(|LTR|, |STR|) ≥ 3n/10

max(|LTR|, |STR|) ≤ 7n/10 ✓



Case 1:

Case 2:

If $k \le |\text{SmallerThanR}|$:

return momselect(k, SmallerThanR) $n' \le 7n/10$

else

return momselect(k − |SmallerThanR|, largerThanR) $n' \le 7$

$$T(n) = T(n/5) + T(7n/10) + O(n)$$

$\boxed{n}$ $O(n)$

$2n/10$ $\boxed{\cancel{n/5}}$ $\boxed{7n/10}$ $\sim 9n/10$

$\sim 81n/100$

$$T(n) = \Theta(n)$$

---

alg with 3 $T(n) = T(n/3) + T(2n/3) + \dots$

n

$(1+\epsilon)n$