

Today: The Start of Divide & Conquer

Contains ($A[0, \dots, n], x$) // Does this array contain the value x .

if the array is empty:

return false

else:

if ($A[0] == x$)

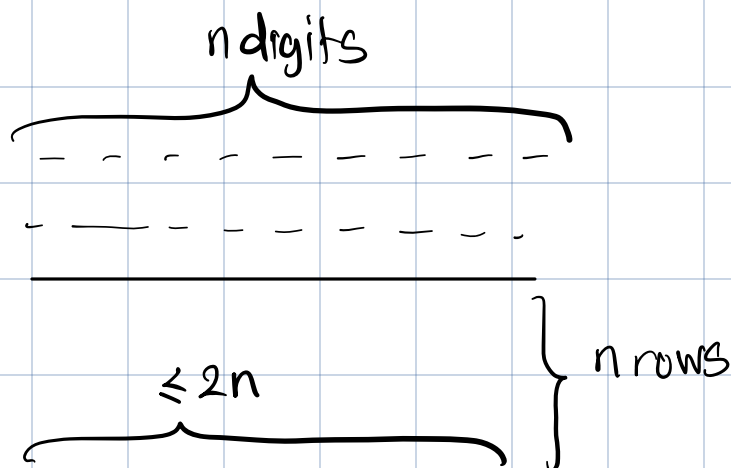
return true

else:

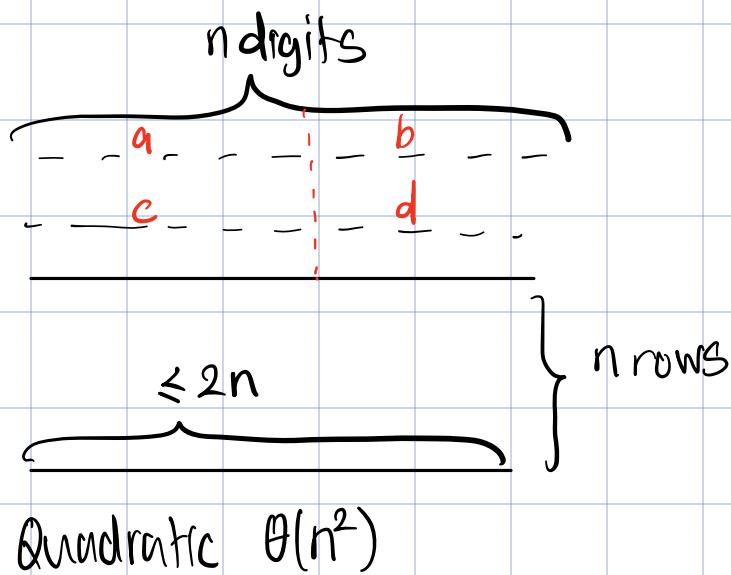
return contains ($A[1:n], x$)

→

1	2	3	4	5	6	7	8
8	7	6	5	4	3	2	1
-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-



Quadratic $\theta(n^2)$



$$\begin{aligned}
 &12345678 \\
 &(1234) \times 10^4 + 5678 \\
 &x = a \times 10^{n/2} + b \\
 &y = c \times 10^{n/2} + d \\
 &(a \cdot 10^{n/2} + b)(c \cdot 10^{n/2} + d) \\
 &= a \times c \times 10^n + (bc + ad) \cdot 10^{n/2} + bd
 \end{aligned}$$

Alg1 (To multiply two n-digit #x & y)

if $n == 1$:

just do it

otherwise:

split into halves, recurse and
put them back together again
as above

Induction before: if

it works for a

particular n , it works

for $n+1$

Induction now: if it

works for all sizes up to

a particular n , it works

for $n+1$.

contains ($A[0, \dots, n], x$) // Does this array contain the value x

if the array is empty: Let $T(n)$ be the runtime of the function

return false

$$T(0) = \theta(1)$$

else:

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

if ($A[0] == x$)

$$\leq \theta(1) + \theta(1) + T(n-2)$$

return true

$$\leq \theta(1) + \theta(1) + \theta(1) + T(n-3)$$

else:

return contains ($A[1:n], x$)

$$= axcx10^n + (bc+ad) \cdot 10^{n/2} + bd$$

Alg1 (To multiply two n -digit x & y)



if $n == 1$:

$$T(1) = \theta(1)$$

just do it

$$T(n) = \theta(1) + 4T(n/2) + \theta(n)$$

Otherwise:

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

split into halves, recurse and

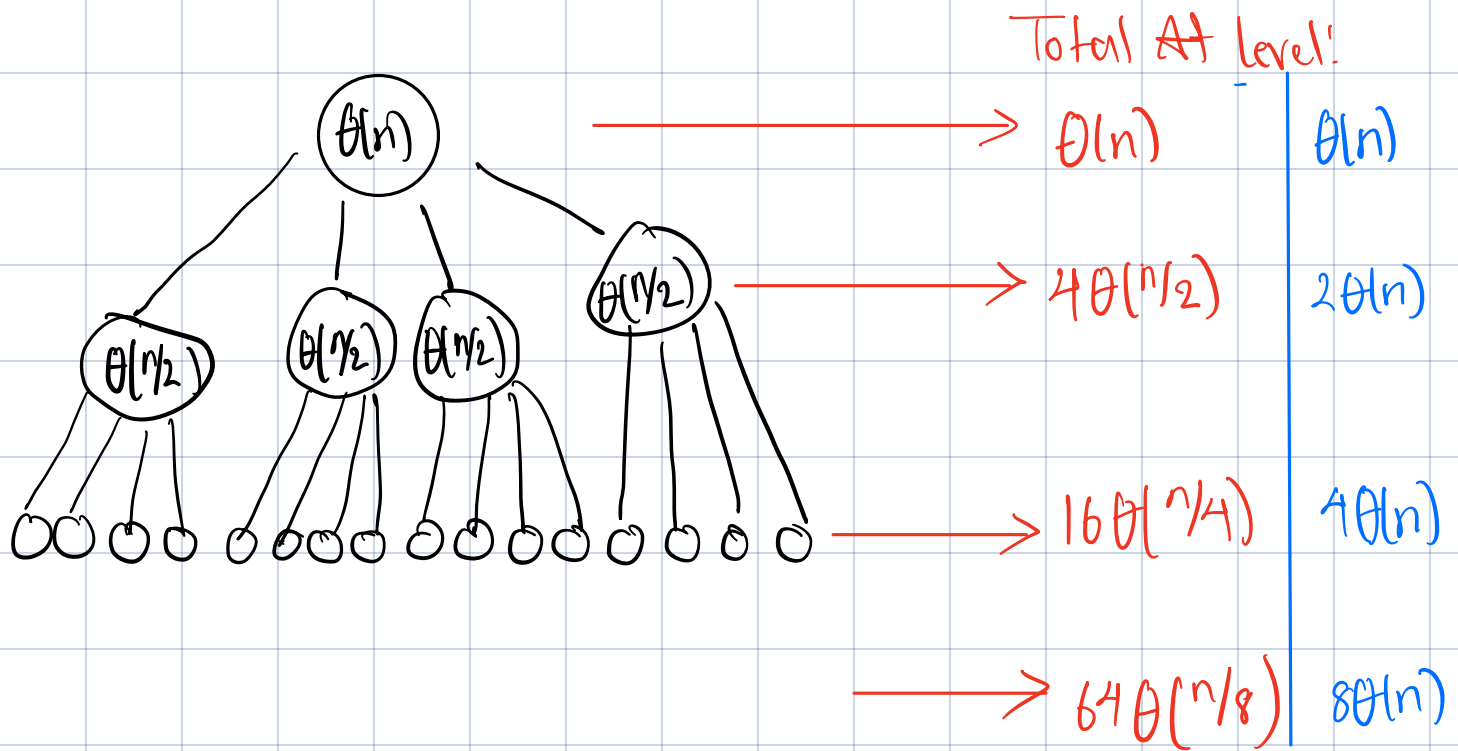
$$= 4(4T(n/4) + \theta(n/2)) + \theta(n)$$

put them back together again

$$= 4(4(4T(n/8) + \theta(n/4)) + \theta(n/2)) + \theta(n)$$

as above

$$\theta(n/2) + \theta(n)$$



Total Work:

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

$$= \sum_{i=0}^{\log_2 n - 1} 2^i \Theta(n)$$

$$= \Theta(n) \sum_{i=0}^{\log_2 n - 1} 2^i$$

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

$$\frac{n}{2^H} = 1$$

$$n^2 = 4$$

$$\log_2 n = \log_2 (2)^H = H$$

$$(a10^{n/2} + b)(c10^{n/2} + d)$$

$$ac10^n + (bc + ad)10^{n/2} + bd$$

$$(a+b)(c+d) = ac + bd + (bc + ad)$$

Karatsuba's Algorithm:

If $n == 1$:

just do it

$$T(1) = \theta(1)$$

Otherwise:

compute $u = a + b$

$v = c + d$

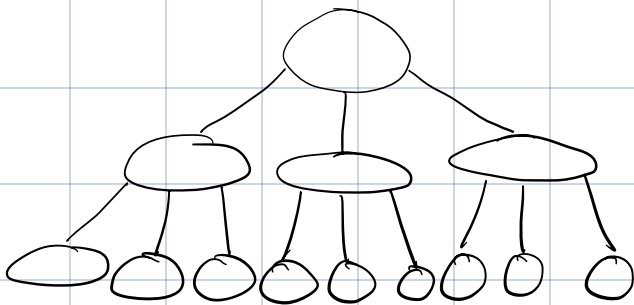
} $\theta(n)$

Recurse on $\begin{matrix} ac \\ bd \\ uv \end{matrix}$

} $3T(n/2)$

compute $bc + ad = uv - (ac + bd)$ $\theta(n)$

put it all together $\theta(n)$



$$\rightarrow \theta(n)$$

$$\rightarrow 3\theta(n/2)$$

$$\rightarrow 9\theta(n/4)$$

Still $\log n$ height

Still work per call

is linear in input size

Total Work

$$\sum_{i=0}^{\log_2 n - 1} \frac{3^i}{2^i} \theta(n) = \sum_{i=0}^{\log_2 n - 1} \left(\frac{3}{2}\right)^i \theta(n)$$

$$\theta(n) \sum_{i=0}^{\log_2 n - 1} \left(\frac{3}{2}\right)^i = \theta(n) \frac{\left(\frac{3}{2}\right)^{\log_2 n} - 1}{\frac{3}{2} - 1}$$

$$= \theta(n) \frac{3^{\log_2 n}}{1}$$

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

$$\sum_{i=0}^N 2^i = 2^{N+1} - 1$$

$$\sum_{i=0}^N r^i = \frac{r^{N+1} - 1}{r - 1}$$

$$\left(\frac{3}{2}\right)^{\log_2 n} = \frac{3^{\log_2 n}}{2^{\log_2 n}} = \frac{3^{\log_2 n}}{n}$$

$$a^{\log_b c}$$

$$\Rightarrow \log_b c = \frac{\log_a c}{\log_a b}$$

$$\frac{3^{\log_2 n}}{n} = \frac{3^{\frac{\log_3 n}{\log_3 2}}}{3^{\frac{n}{\log_3 2}}}$$

$$(a^b)^c = a^{bc} \quad \downarrow \downarrow \downarrow$$

$$= \left(3^{\log_3 n} \right)^{\log_3 2}$$

$$= n^{\log_2 3} \approx n^{1.58}$$