mergesort (array A of length n):
    if $n \le 1$:
        return A
    else:
    x = mergesort (L half of A)
    y = mergesort (R half of A)
    return (merge (x, y))

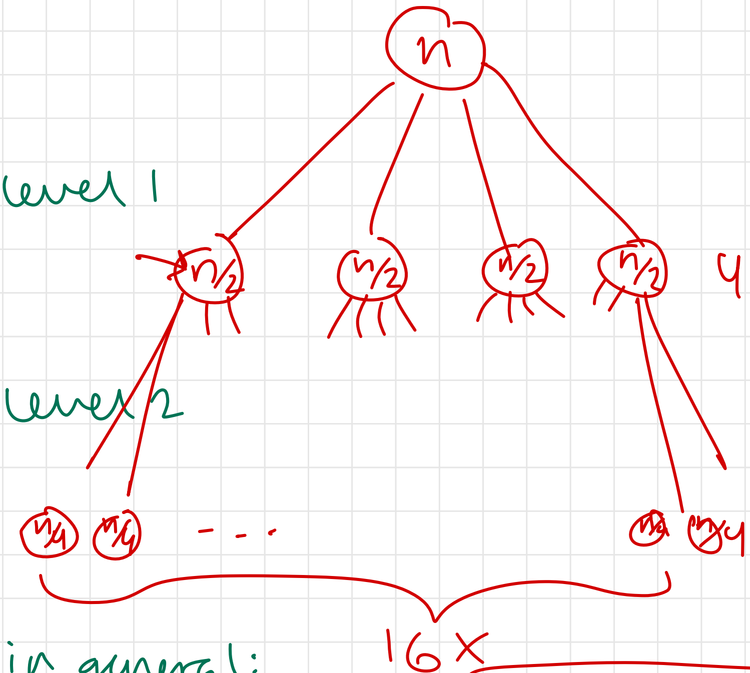
rec. calls ← non-recursive work

$$T(n) = \underset{a}{4}\,T\left(\frac{n}{\underset{b}{2}}\right) + n \quad , \quad T(1) = 1$$

level 0

sum

$1 \cdot n = n$



level 1

sum

$4 \cdot \frac{n}{2} = 2n$

level 2

sum

$4n = 16 \cdot \frac{n}{4}$

$n\left(\frac{a}{b}\right)^{\ell}$

in general:

$n \cdot 2^{\ell}$

$$\boxed{T(n) = aT\left(\frac{n}{b}\right) + n}$$

work is: increasing
decreasing
same

levels of tree: $\log_2(n)$

$\log_b(n)$

Overall runtime:

\# children a node has ^ height of tree

because that's # leaves · 1

plug # levels into general work / level formula

multiply(x, y, n)
    if n=1:
        return $x \cdot y$
    else:
        # get smaller numbers (fewer digits than n)
        #
        # call multiply on those
        # use result to get $x \cdot y$

        $m = n/2$

        a = first half of digits of x
        b = second half
        c = first half of digits of y
        d = second half

n

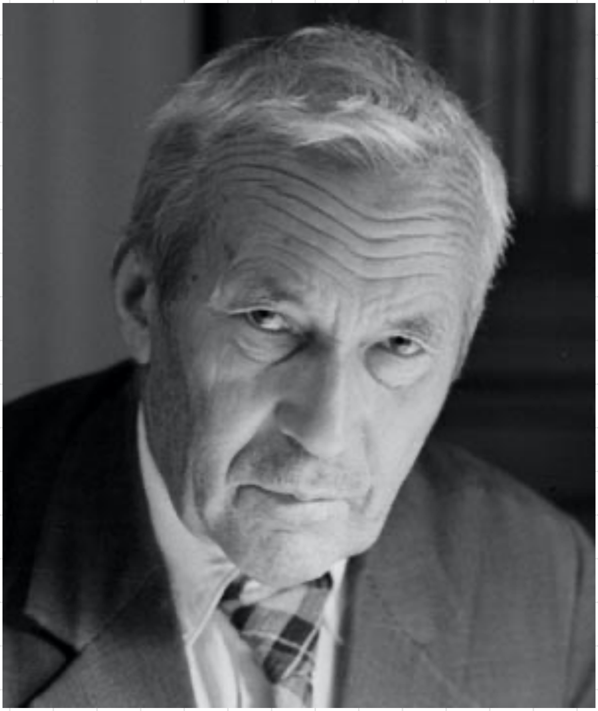
        e = multiply(a, c, m)
        f = multiply(b, d, m)
        g = multiply(b, c, m)
        h = multiply(a, d, m)

        return $e \cdot \underbrace{10^{2m}}_{n} + (g+h)\underbrace{10^{m}}_{n} + f$

$T(n) = 4\left(T\left(\frac{n}{2}\right)\right) + n$, so $\Theta(n^2)$

$x \cdot y = (a\,10^m + b)(c\,10^m + d)$

$= (\overset{e}{a \cdot c})\,10^{2m} + (\overset{g}{b \cdot c} + \overset{h}{a \cdot d})\,10^m + \overset{f}{b \cdot d}$



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