Divide + Conquer (cont)

- TAs have office hours!
- 1 section (optional) on Fri (TBD)
- Youtube videos to explain basics.
- HW 1 due this Thursday

Master Thm

break a problem of size n into

a subproblems of size $\frac{n}{b}$ +

it takes n^d steps to merge

so $$T(n) = a\,T\left(\frac{n}{b}\right) + n^d$$

then:

$$T(n) = \begin{cases} O(n^d) & \text{if } a < b^d \\ O(n^d \log n) & \text{if } a = b^d \\ O(n^{\log_b a}) & \text{if } a > b^d \end{cases}$$

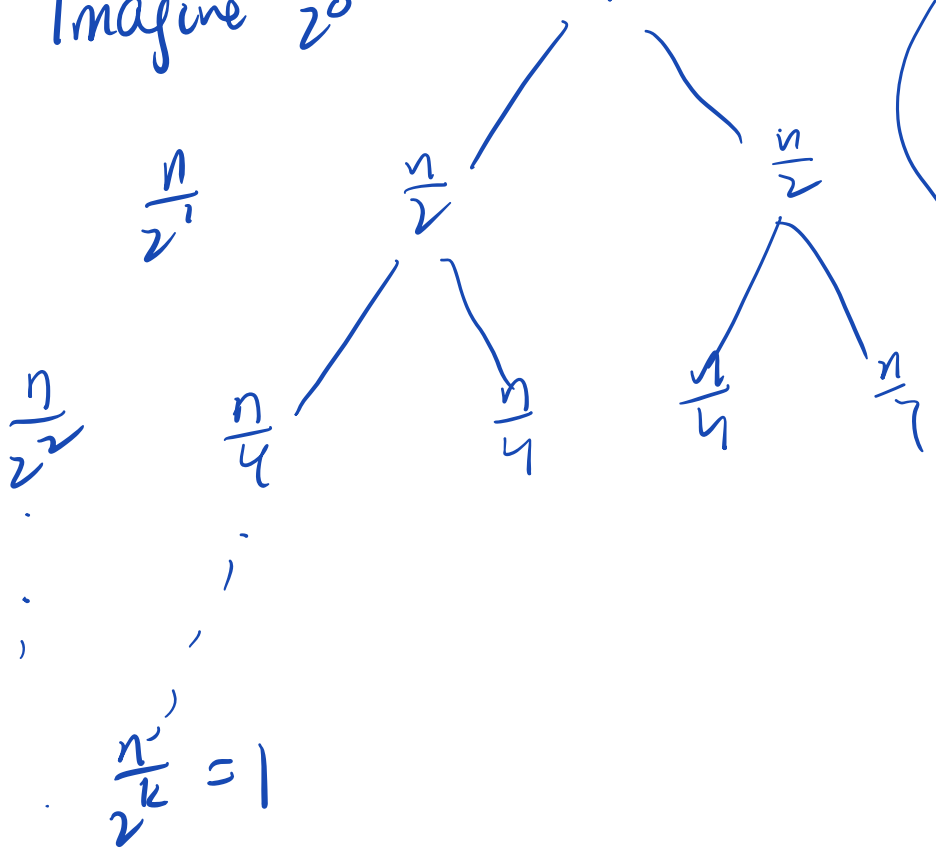
Ex: Binary search

$\log_2(n) = O(\log_{1000} n)$

and so forth

so big-O tells you ignore base

Imagine $\frac{n}{2^0} = n$



Let $n = 2^k$

$k = \log_2 n$

Binary search

Alg: Check the middle + look

at one side other + repeat

so $T(n) = T\left(\frac{n}{2}\right) + O(1)$

$a = 1$

$b = 2$

$d = 0$

Compare 1 and 2^0

a b^d

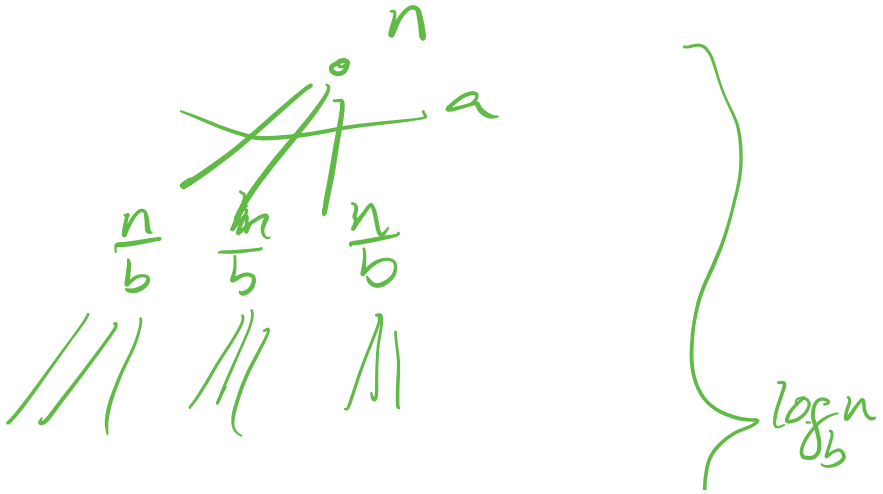
same

$\Rightarrow T(n) = O(n^d \log n)$

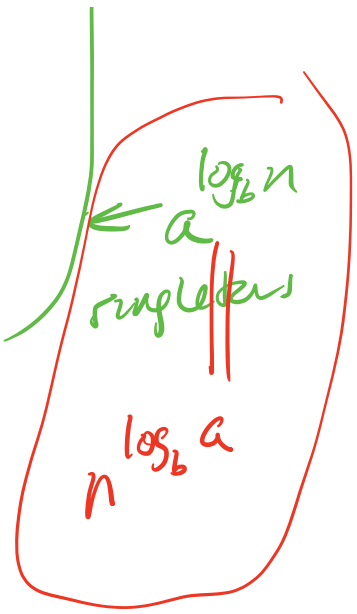
$= O(\log n)$ if $d = 0$.

Mathematician reason

merge takes n^d →



merge → a ##



Ex 2: What about

binary search + O(1)

$$T(n) = T\left(\frac{n}{2}\right) + O(n)$$

$a = 1$

$b = 2$

$d = 1$

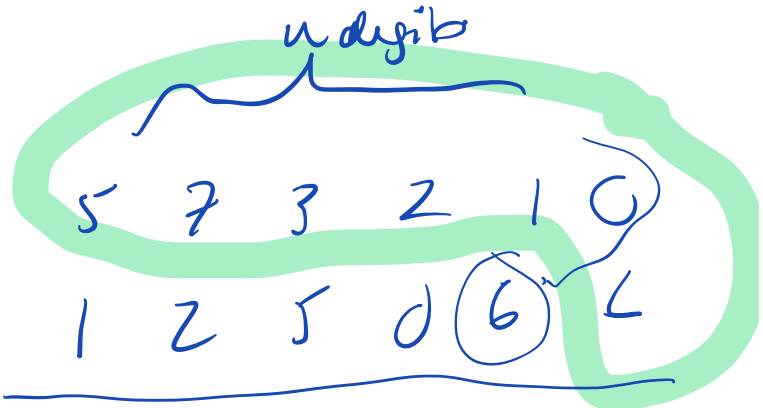
compare

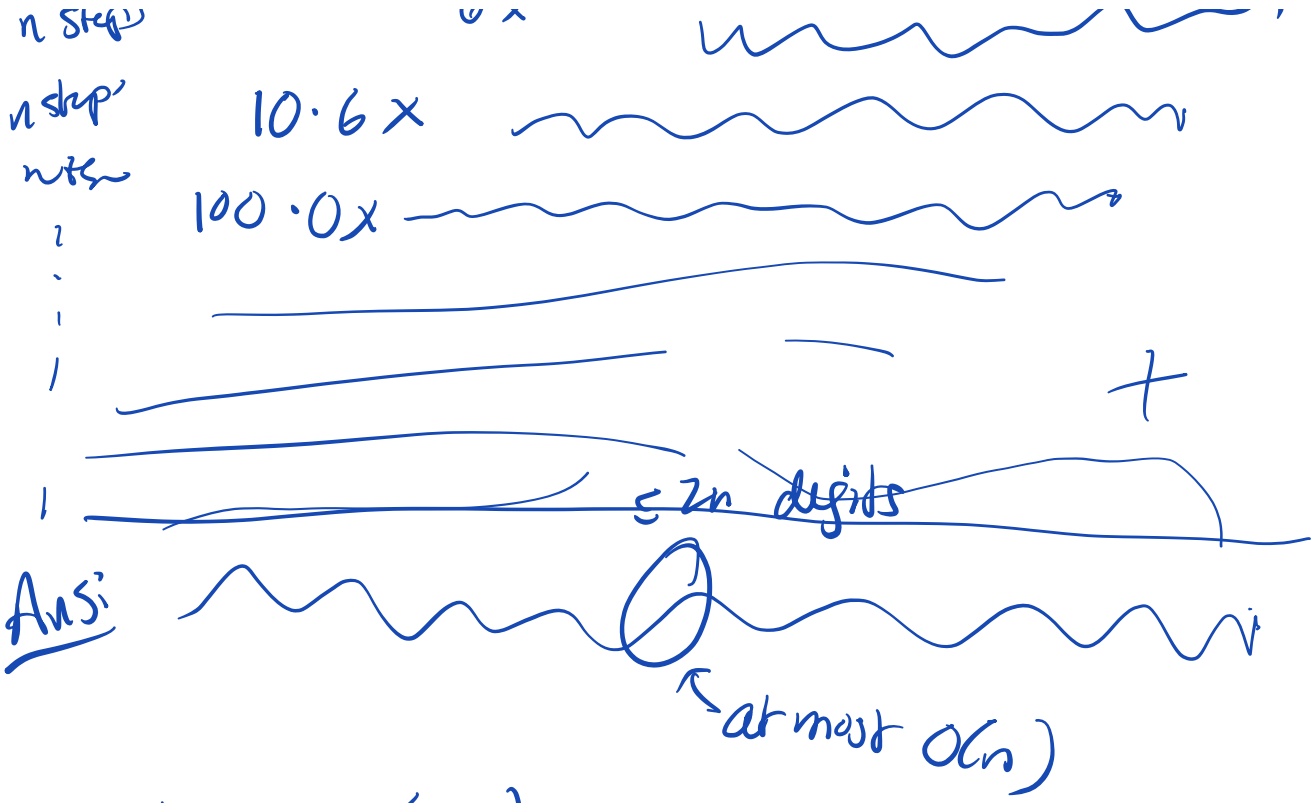
$1 < 2^1$

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

Multiplication

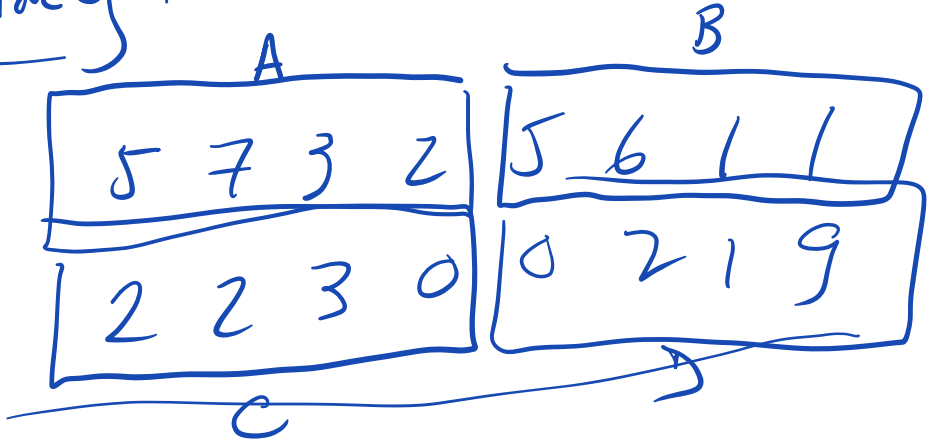
5th grade





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

Alternatively:



A, B, C, D are all $\frac{n}{2}$ digit #s.

Observe: n) / $\frac{n}{2}$)

Goal: $(A \cdot 10^{\frac{n}{2}} + B)(C 10^{\frac{n}{2}} + D)$

$= AC \cdot 10^{n} + (AD + BC) 10^{\frac{n}{2}} + BD$

must 2 $\frac{n}{2}$ digit #'s

adding takes $O(n)$ steps

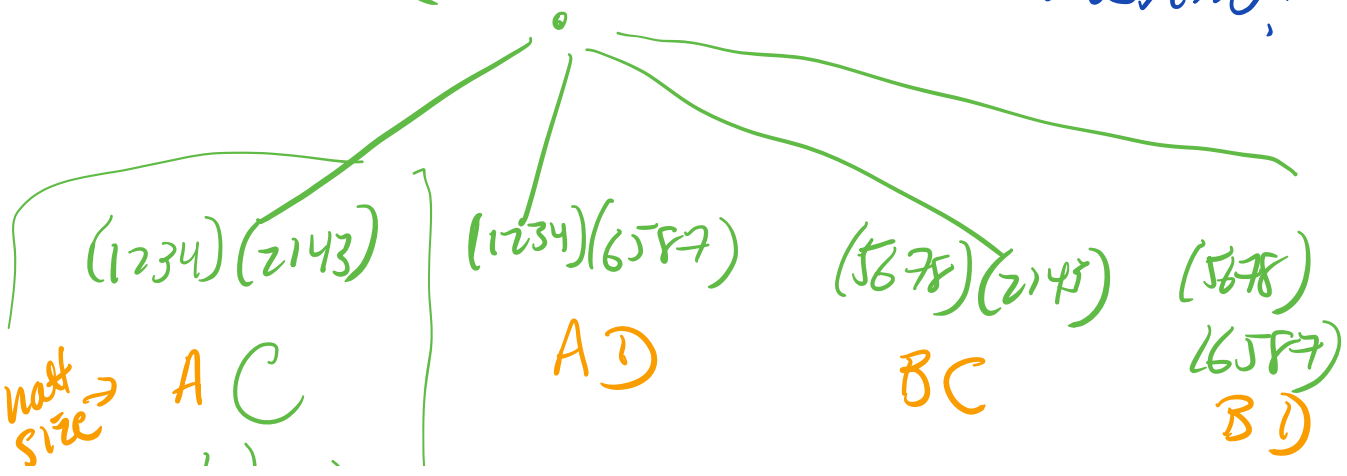
$a = 4$
$b = 2$
$d = 1$

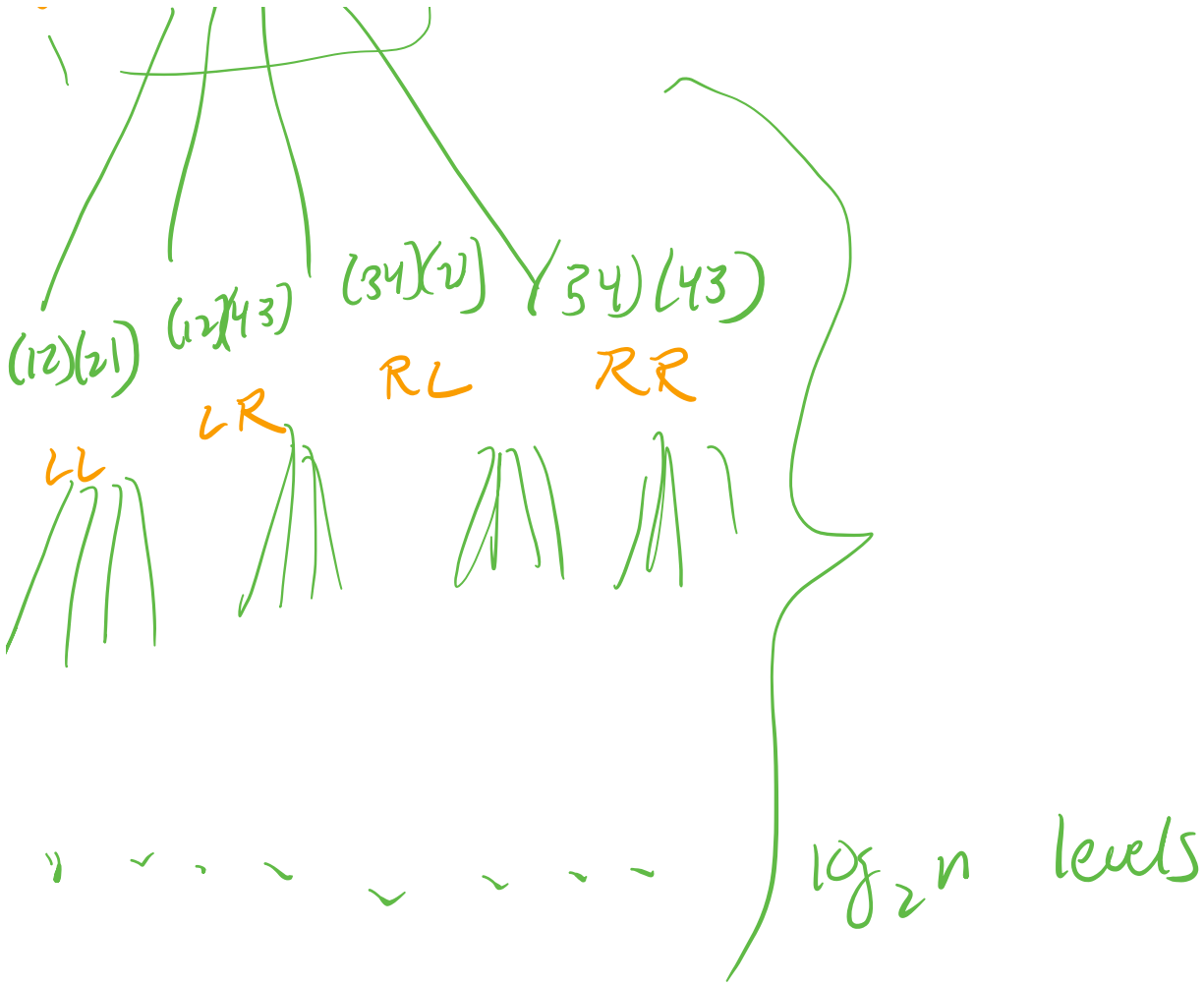
Easy: $T(n) = 4\,T(\frac{n}{2}) + O(n)$

Solve: compare $a = 4 > b^d = 2^1$

$O(n^{\log_b a}) = O(n^{\log_2 4}) = O(n^2)$

(12345678)(21436587) AGAIN!



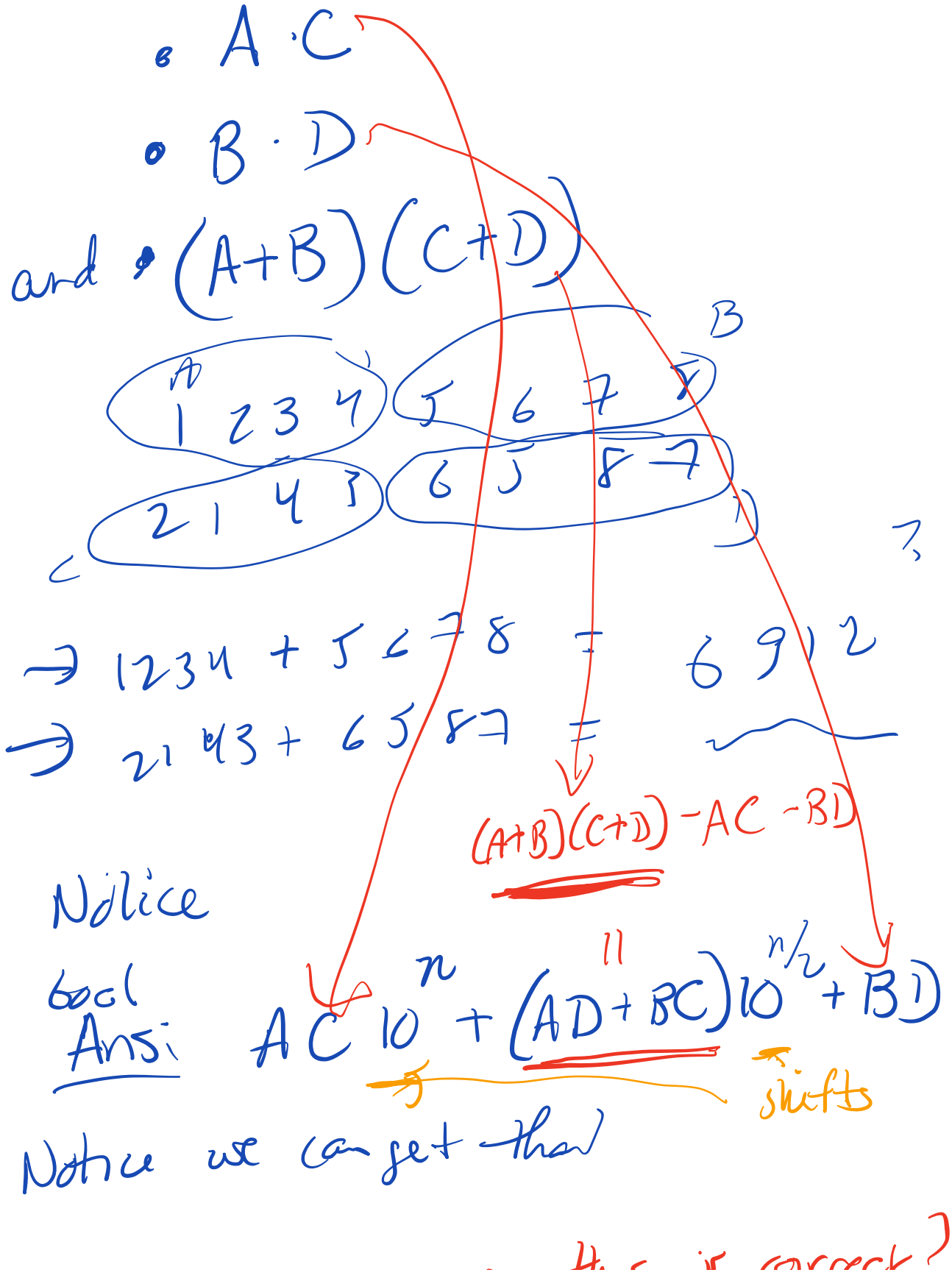


Karatsuba

Mult $X = 10^{\frac{n}{2}} A + B$

$Y = 10^{\frac{n}{2}} C + D$

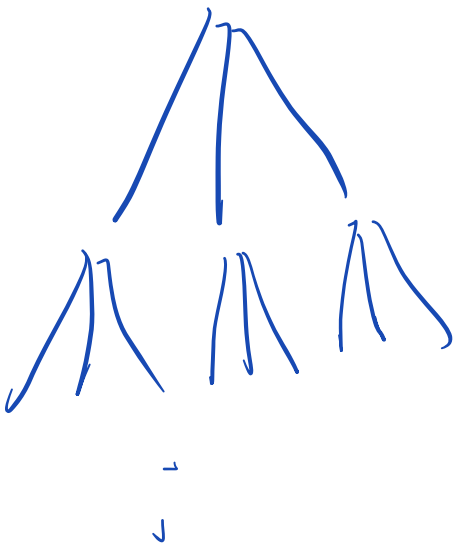
Instead calculate


A·C
B·D
and (A+B)(C+D)
A
B
C
D
1 2 3 4
5 6 7 8
2 1 4 3
6 5 8 7
1234 + 5678 = 6912
2143 + 6587 =
(A+B)(C+D) - AC - BD
Nice
Ans:
AC 10^n + (AD+BC) 10^(n/2) + BD
shifts
Notice we can get that

1. Do we agree this [illegible]

Harder

$$T(n) = 3\,T\left(\frac{n}{2}\right) + O(n)$$



instead

Compare

$$a \quad ? \quad b^d$$

$$3 \quad > \quad 2^1$$

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

$$= O(n^{\log_2 3})$$

$= O(n^{1.59})$

Median finding

Unordered list, want middle value