

Divide & Conquer

- ① Break problem into subproblems. ←
 - ② Recursively solve subproblems. ←
 - ③ Combine answers into overall answer. ←
-

Integer multiplication

$$A = 2^{n/2} A_1 + A_2$$

$$B = 2^{n/2} B_1 + B_2$$

n-bit product.

$$AB = (2^{n/2} A_1 + A_2)(2^{n/2} B_1 + B_2)$$

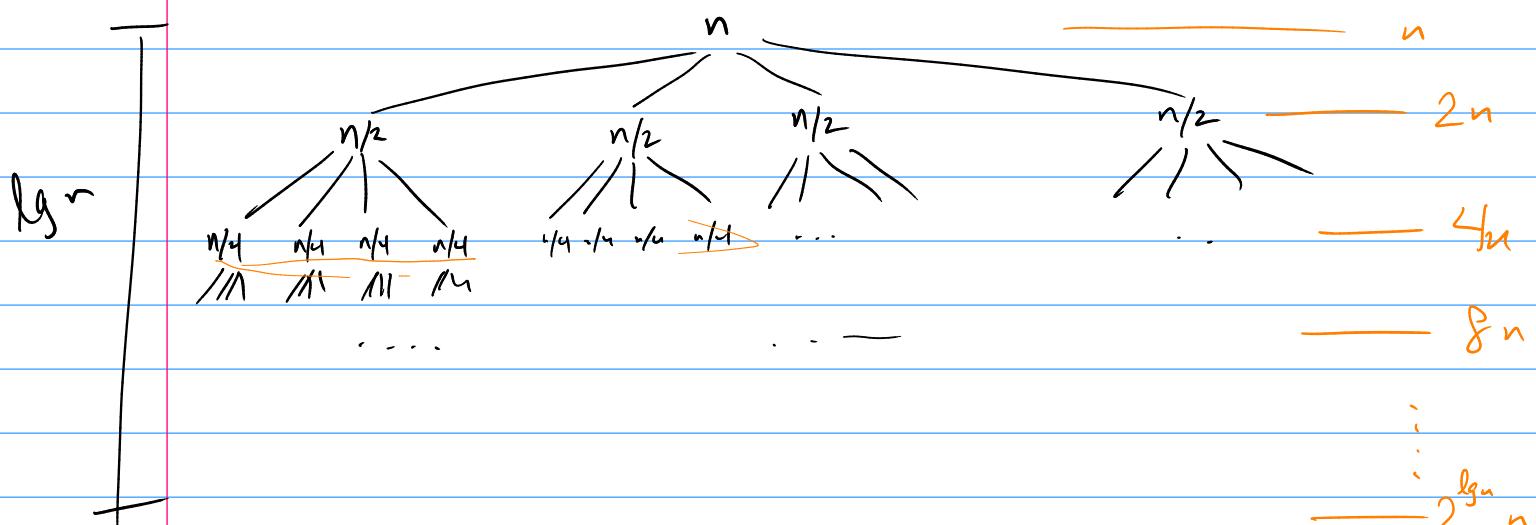
$$= 2^n A_1 B_1 \underset{\substack{+ \\ \text{n/2-products}}}{} + 2^{n/2} (A_1 B_2 + A_2 B_1) \underset{\substack{+ \\ \text{3 additions}}}{} + A_2 B_2$$

2 shifts 3 additions

$$M(1) = \Theta(1)$$

$$M(n) = 4M(n/2) + \underline{\Theta(n)}$$

Can we come up w/ closed form for M ?



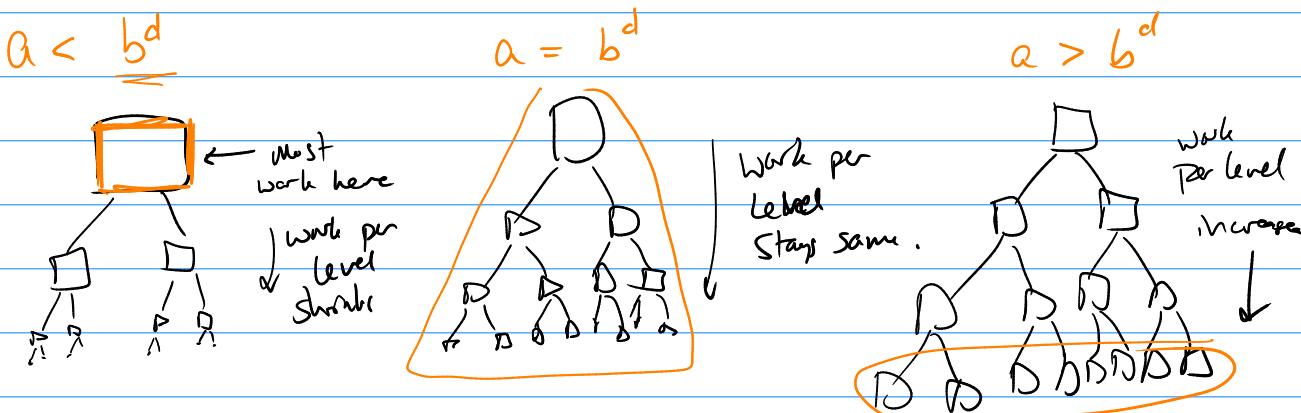
$$\begin{aligned} \text{Total work} &= n + 2n + 4n + 8n + \dots + n^2 \\ &= n(1 + 2 + 4 + 8 + \dots + n) \\ &= \Theta(n^2) \quad !! \end{aligned}$$

Theorem. (Master Theorem)

If $T(n) \leq aT(n/b) + O(n^d)$

for positive constants a, b, d , then

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



Examples

- Merge sort : $a = 2, b = 2, d = 1$.

$$a = 2 \quad b^d = 2^1 = 2.$$

Hence $a = b^d$ so merge sort takes $O(n^d \lg n) = O(n \lg n)$. ✓

- Binary search : $a = 1, b = 2, d = 0$ ($B(n) = B(n/2) + O(1)$)

$$a = 1 = b^d = 2^0 = 1.$$

Hence binary search takes $O(n^d \lg n) = O(\lg n)$.

- Recursive n -bit multiplication : $a = 4, b = 2, d = 1$

$$4 > 2^1$$

Hence it takes $O(n^{\log_b a}) = O(n^{\log_2 4}) = O(n^2)$.

Karatsuba multiplication

$$A = 2^{n/2} A_1 + A_2$$

$$B = 2^{n/2} B_1 + B_2$$

$$P_1 = A_1 B_1$$

$$P_2 = A_2 B_2$$

$$P_3 = (A_1 + A_2)(B_1 + B_2)$$

$$\text{Notice } P_3 - P_1 - P_2 = \underline{\underline{A_1 B_2 + A_2 B_1}}$$

$$\begin{aligned} \text{Hence } AB &= 2^n A_1 B_1 + 2^{n/2}(A_1 B_2 + A_2 B_1) + A_2 B_2 \\ &= 2^n P_1 + 2^{n/2}(P_3 - P_1 - P_2) + P_2. \end{aligned}$$

Only 3 multiplications instead of 4! (But
more additions).

So $a = 3$, $b = 2$, $d = 1$

still $\Theta(n)$ work even though there are more additions.

$a > b^d$, so it takes

$$\mathcal{O}(n^{\log_b a}) = \mathcal{O}(n^{\log_2 3}) \approx \mathcal{O}(\underline{\underline{n^{1.59...}}})$$