Divide + Conquer algorithms

1. Split up problem into subproblems
2. Recursively solve each subproblem
3. Combine answers.

Nice when it works:

- Implementation: recursion
- Proof of correctness: induction.
- Asymptotic analysis: Master Theorem.

Monday: integer multiplication.

$$A = A_1 2^{n/2} + A_2$$
$$B = B_1 2^{n/2} + B_2$$

e.g. $A = \overbrace{0110}^{A_1}\overbrace{1011}^{A_2}$

$$AB = (A_1 2^{n/2} + A_2)(B_1 2^{n/2} + B_2)$$
$$= A_1B_1 2^n + (A_1B_2 + A_2B_1)2^{n/2} + A_2B_2$$

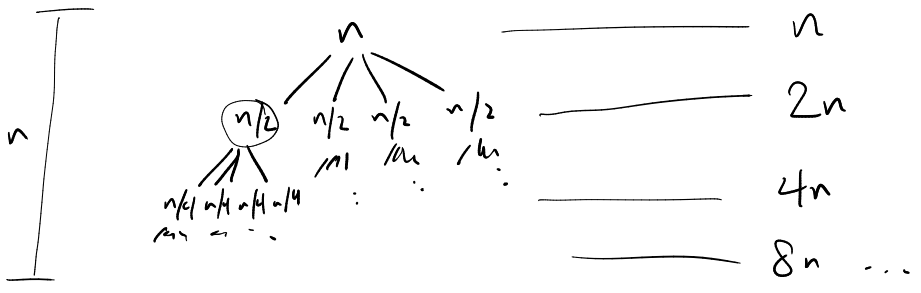
2 bit-shifts

3 n-bit additions.

4 $n/2$-bit multiplications.

$M(n)$ = time to multiply 2 n-bit #'s by the above recursive algorithm.

$$M(1) = O(1)$$
$$M(n) = 4M(n/2) + O(n).$$



$$\text{Total} = n + 2n + 4n + 8n + \dots + 2^k n + \dots + 2^{\lg n} n$$

$$\leq n + 2n + 4n + 8n + \dots + n \cdot n$$

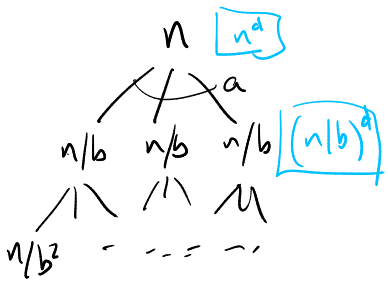
$$= n(1 + 2 + 4 + 8 + \dots + n)$$

$$\leq n(2n - 1)$$

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

$$\left[1 + 2 + 4 + \dots + 2^n = 2^{n+1} - 1 \right]$$

In general: $T(n) = aT(n/b) + O(n^d)$



n^d

$a \cdot (n/b)^d$

$a^2 (n/b^2)^d$

$\vdots$

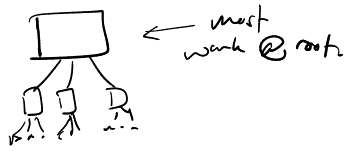
$a^k (n/b^k)^d$

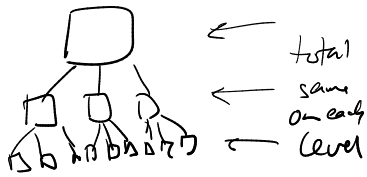
$= n^d \left(\frac{a}{b^d}\right)^k$

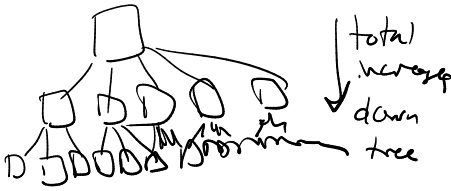
Master Theorem. Suppose $a, b > 0$ and $d \geq 0$ and

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

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







eg. Merge sort. $a = 2 \quad b = 2 \quad d = 1$

$2 = 2^1$, hence Case 2

$\rightarrow O(n^d \lg n) = O(n \lg n)$

& Binary search. $a = 1 \quad b = 2 \quad d = 0$.

$a \quad b^d$

$1 = 2^0 = 1$, Case 2.

$\rightarrow O(n^d \lg n) = O(\lg n)$.

eg. Multiplication from activity:

$a = 4, \; b = 2, \; d = 1.$ $\quad 4 > 2^1$, so Case 3

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

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

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

$P_1 = A_1 B_1$ — 3 $n/2$-bit multiplications

$P_2 = A_2 B_2$

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

$= A_1 B_1 + A_1 B_2 + A_2 B_1 + A_2 B_2$

Hence $P_3 - P_1 - P_2 = A_1 B_2 + A_2 B_1$

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

$= P_1 2^n + (P_3 - P_1 - P_2) 2^{n/2} + P_2.$

6? additions — still $O(n)$

Hence $M(n) = 3M(n/2) + O(n)$

$a = 3, \; b = 2, \; d = 1 \quad . \quad 3 > 2^1$, Case 3.

Hence $M(n) = O(n^{\log_b a}) = O(n^{\log_2 3}) \approx O(n^{1.59})$