Divide and Conquer [chapter 5]

- What are some of algorithms that used the divide and Conquer?
    - Binary Search
    - Merge Sort
    - Integer Multiplication.

How many single digit multiplications were done?

$$
\begin{array}{r}
2311 \\
\times \ 3234 \\
\hline
9244 \\
6933 \\
4622 \\
6933 \\
\hline
7473774
\end{array}
$$

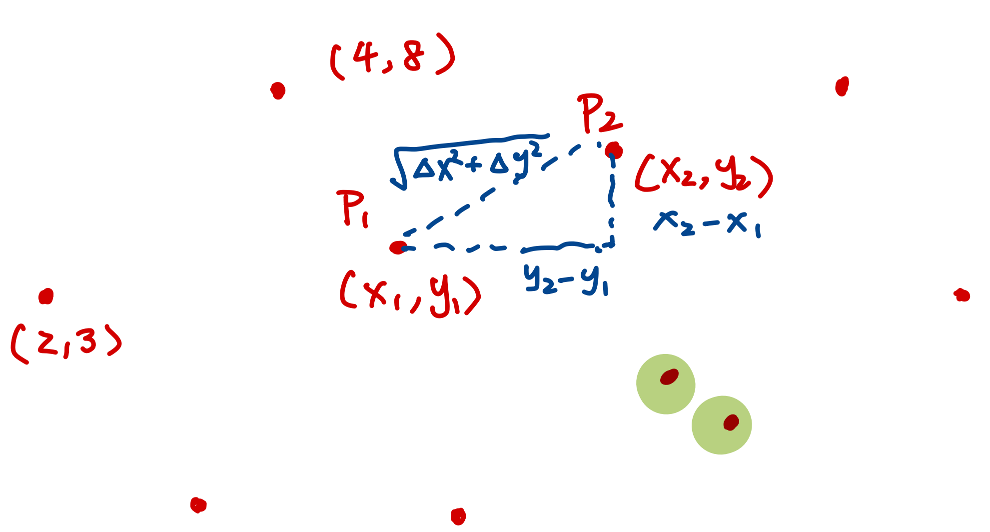
16 multiplications

$$
\begin{array}{rcccc}
 & a_1 & a_2 & \cdots & a_n \\
\times & b_1 & b_2 & \cdots & b_n \\
\hline
\end{array}
\qquad \boxed{n^2}
$$

Using D&C, we can improve runtime to $n^{1.59}$.

- Closest Pair of Points
    - Given a set of n points (x_k, y_k)

    Question: Return the closest pairs of points.



$$d(P_1, P_2) = \sqrt{(x_2-x_1)^2 + (y_2-y_1)^2}$$

Naive: Check every pairs

$\binom{n}{2}$ pairs $= O(n^2)$

Use D&C, we can be done in $O(n \log n)$.

- Review Recurrence Relation:

- Find $f(n)$, $f(n) = f(n/2) + 10$, $n = 2^k$, $f(1) = 1$.

$$f(2^k) = f(2^{k-1}) + 10 \qquad n = 2^k \rightarrow k = \log_2 n$$

$$f(2^{k-1}) = f(2^{k-2}) + 10 \qquad n = 2^{k-1}$$

$\vdots$

$n = 2^4 \quad f(16) = 41$

$n = 2^3 \quad f(8) = 31$

$n = 2^2 \quad f(4) = f(2) + 10 = 21$

$n = 2^1 \quad f(2) = f(1) + 10$

$= 1 + 10 = 11$

$$\boxed{\begin{aligned} f(2^k) &= 10k + 1 \\ f(n) &= 10 \log_2 n + 1 \end{aligned}}$$

- $f(n) = f(n/2) + n$, $f(1) = 1$. $\quad f(n) = 2^{\log_2 n + 1} - 1$

$n = 2^k$, $k = \log_2 n$ $\qquad f(2^k) = \sum_{i=0}^{k} 2^i$

$$f(2^k) = f(2^{k-1}) + 2^k$$

$$f(2^{k-1}) = f(2^{k-2}) + 2^{k-1}$$

$\vdots$

$f(8) = 1 + 2 + 4 + 8$

$f(4) = f(2) + 4 = 1 + 2 + 4$

$f(2) = f(1) + 2 = 1 + 2$

$S = 1 + 2 + 4 + 8 + \cdots + 2^k$

$= 2^{k+1} - 1$

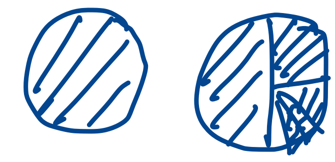
$2S = \underbrace{2 + 4 + 8 + \cdots + 2^k} + 2^{k+1}$

$2S = S - 1 + 2^{k+1}$

$-S \qquad -S$

$S = 2^{k+1} - 1$

- Geometric series summation formulas:



- $1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \cdots = 2$

- $a < 1$, $\underbrace{1 + a + a^2 + a^3 + \cdots}_{\text{call it } S.} = \frac{1}{1-a}$

$$aS = a + a^2 + a^3 + \cdots$$

$$aS = S - 1$$

$$1 = S - aS$$

$$1 = S(1-a)$$

$$S = \frac{1}{1-a}$$

$a = \frac{1}{2}$

$$\frac{1}{1-\frac{1}{2}} = \frac{1}{\frac{1}{2}} = \boxed{2}$$

- $a > 1$, $\underbrace{1 + a + a^2 + \cdots + a^n}_{\text{call it } S.} =$

$$aS = \underbrace{a + a^2 + a^3 + \cdots + a^n} + a^{n+1}$$

$$aS = S - 1 + a^{n+1}$$

$$-S \qquad -S$$

$$S(a-1) = aS - S = a^{n+1} - 1$$

$$S = \frac{a^{n+1} - 1}{a - 1}$$

Ex: $a = 2$

$$\frac{2^{n+1} - 1}{2 - 1} = 2^{n+1} - 1$$

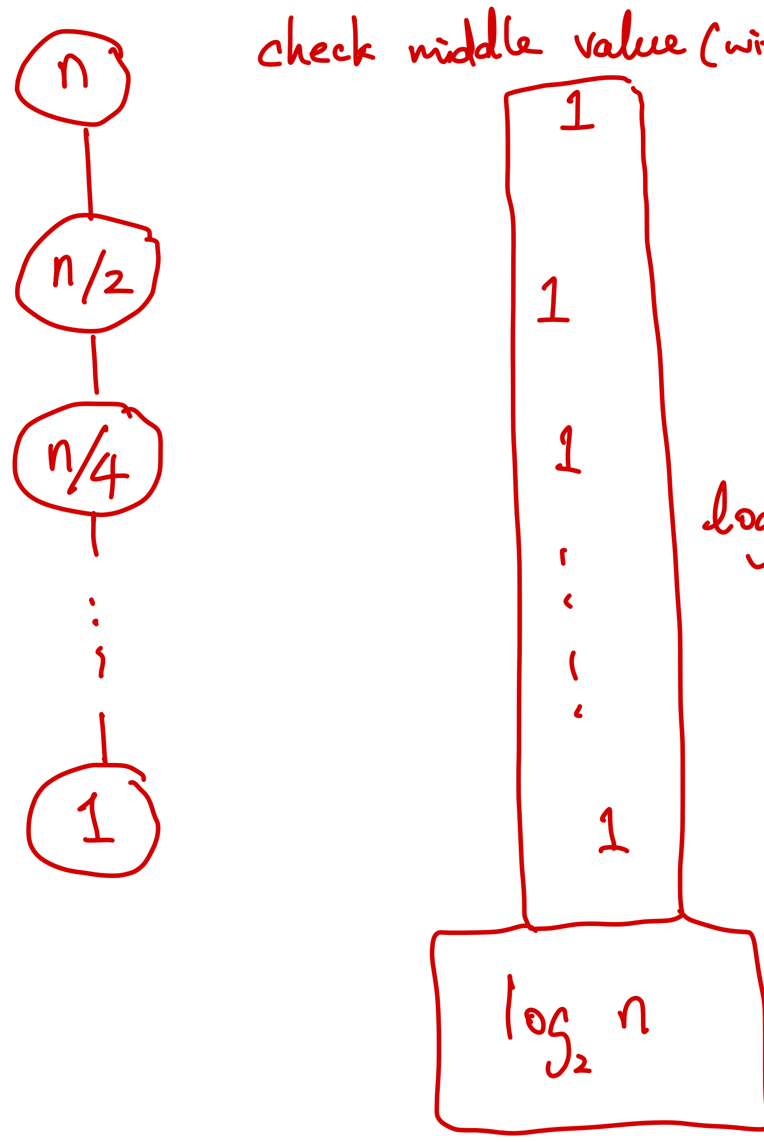
- $f(n) = f(n/2) + 1$

  If $f(n)$ represents runtime of an algorithm on input size n.

  Which algorithm can this formula represent?

Binary.

check middle value (with target)



$f(n) = \log_2 n$

$f(n) = f(\frac{n}{2}) + 1$

$f(\frac{n}{2}) = f(\frac{n}{4}) + 1$

$\vdots$

$f(2) = f(1) + 1 = 2$

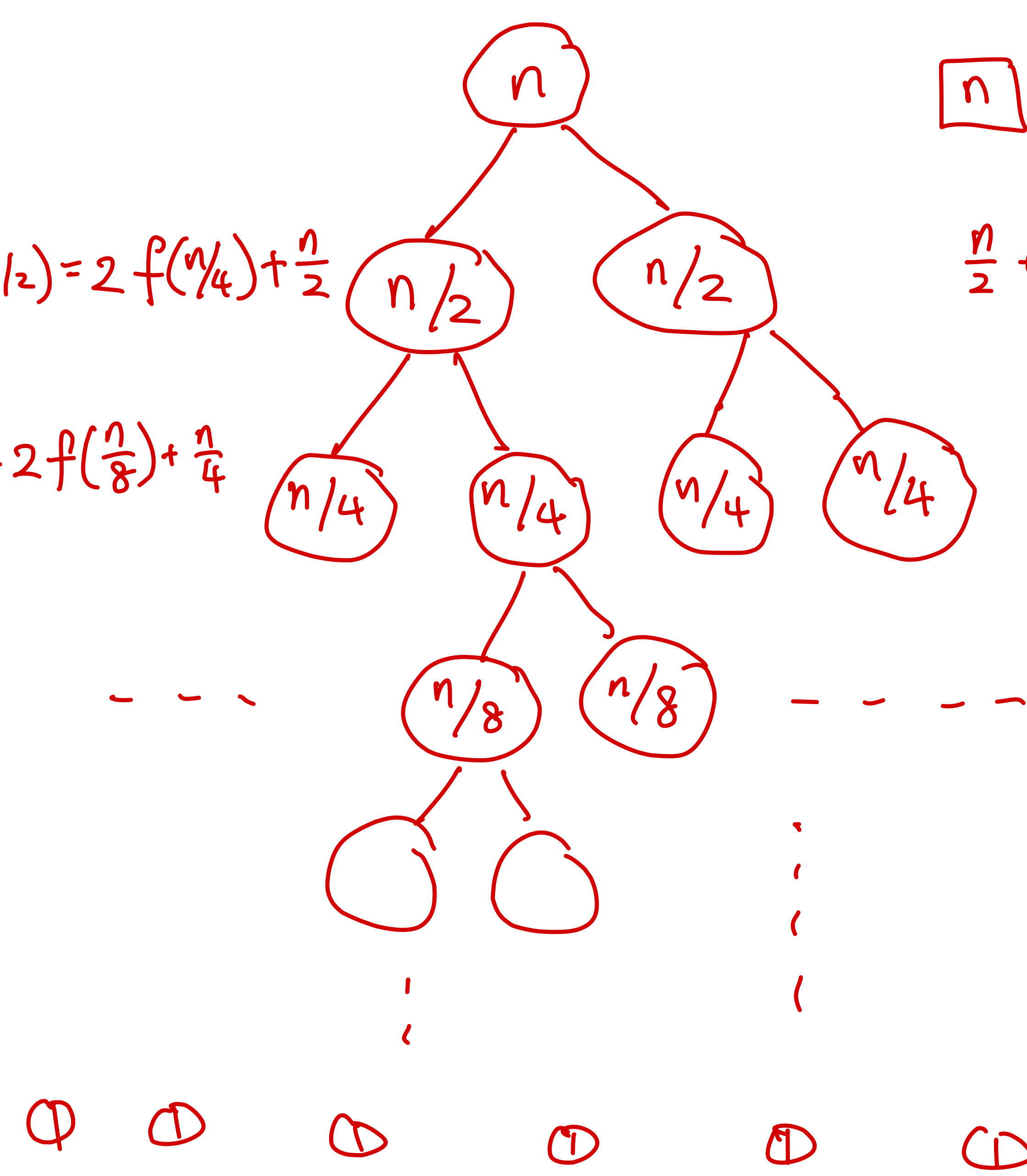
$f(1) = 1$

Total: $\boxed{n \cdot \log n}$

- $f(n) = 2f(n/2) + n$

$f(n/2) = 2f(n/4) + \frac{n}{2}$

$f(\frac{n}{4}) = 2f(\frac{n}{8}) + \frac{n}{4}$



$\boxed{n}$ $\boxed{2n}$

$\frac{n}{2} + \frac{n}{2} = n$ $\boxed{2n}$

$\frac{n}{4} + \frac{n}{4} + \frac{n}{4} + \frac{n}{4} = n$ $\boxed{2n}$

$\frac{n}{8} * 8 = n$ $\boxed{2n}$

$\vdots$

n $\boxed{2n}$

$\log_2 n$

Divide & Conquer

Given a problem P (size n)

Divide P into subproblems (with size $\frac{n}{B}$)

Subproblems are solved recursively.

Merge the solutions to solve P.

$T(n)$ - time taken by algorithm on input size n.

$$T(n) = A \cdot T\left(\frac{n}{B}\right) + f(n)$$

number of subproblems

size of each subproblems

recombining merge step cost.

- Merge Sort $O(n \log n)$ Sorting algorithm.

(Optimal in the comparison model).

Original list / array A of length n.

$A[1, 2, \ldots, n]$

$T(n)$

Sort (A):

if $n = 1$, return $A[1]$ ← Base case. 1

$L = A[1, 2, \ldots \frac{n}{2}]$ $n/2$

$R = A[\frac{n}{2}+1, \ldots, n]$ $n/2$

n

$$\boxed{T(n) = 2T(n/2) + 2n}$$

$LS = \text{Sort}(L)$ $T(n/2)$

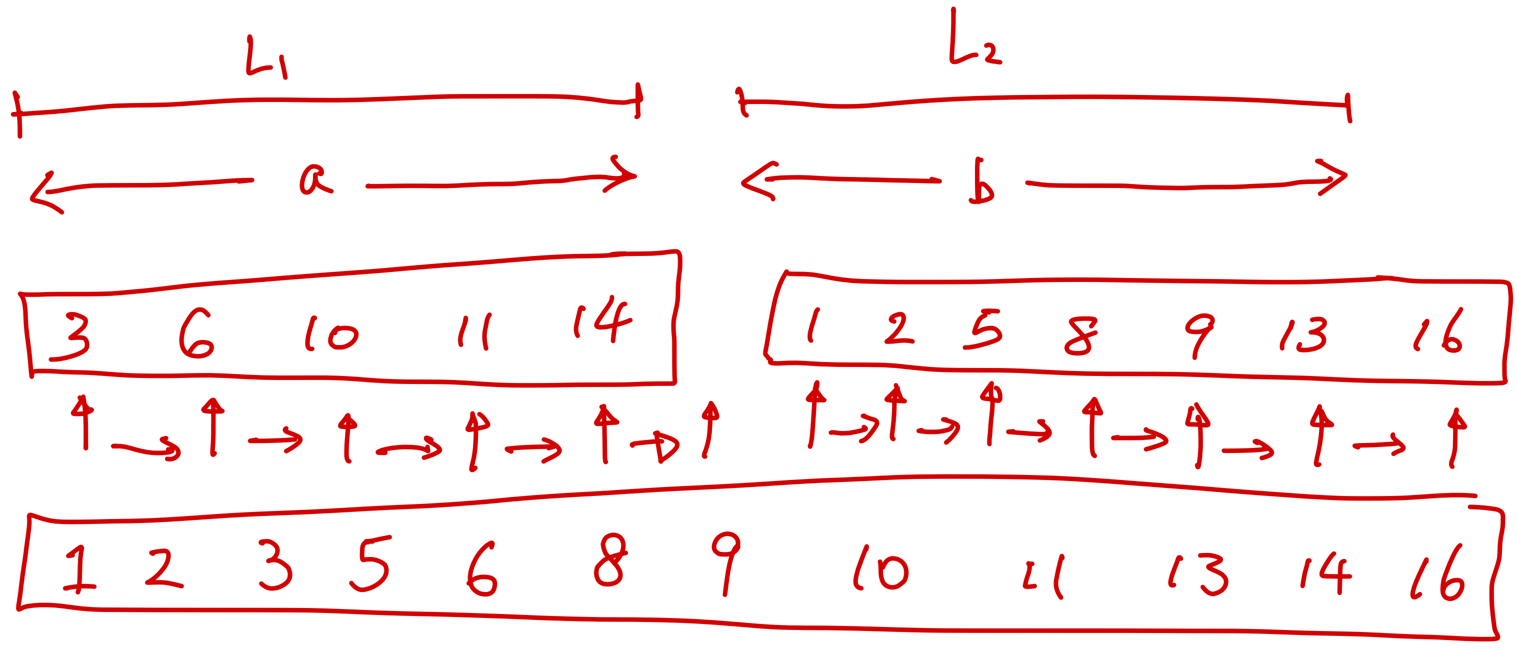
$RS = \text{Sort}(R)$ $T(n/2)$

$$\boxed{T(n) = 2n \log_2 n = O(n \log n)}$$

Output merge (LS, RS). $n-1$

takes two sorted lists and merge them.

merge (L_1, L_2) Two pointers algorithm



$a+b-1$