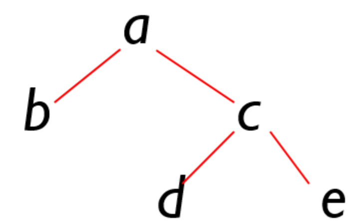
Divide-and-Conquer

Binary Tree Algorithms

Binary tree is a divide-and-conquer ready structure!

Classic traversals (pre-order, in-order, post-order)



$Algorithm\ Inorder(T)$

$if\ T \neq \emptyset$

$Inorder(T_{left})$

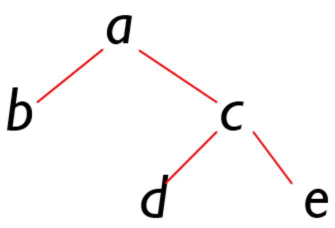
$print(root\ of\ T)$

$Inorder(T_{right})$

Binary Tree Algorithms

Binary tree is a divide-and-conquer ready structure!

Classic traversals (pre-order, in-order, post-order)



$$Algorithm\ Inorder(T)$$

$$if\ T \neq \emptyset$$

$$Inorder(T_{left})$$

$$print(root\ of\ T)$$

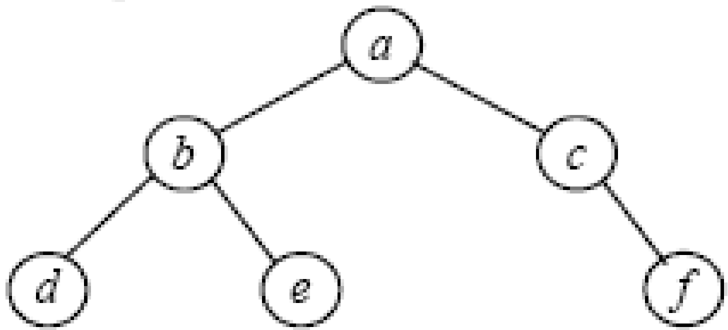
$$Inorder(T_{right})$$

- Efficiency: $\Theta(n)$

Discussion

Traverse the following binary tree

a.. in preorder. b. in inorder. c. in postorder.



$Algorithm\ Preorder(T)$
$if\ T \neq \emptyset$
$\quad print(root\ of\ T)$
$\quad Preorder(T_{left})$
$\quad Preorder(T_{right})$

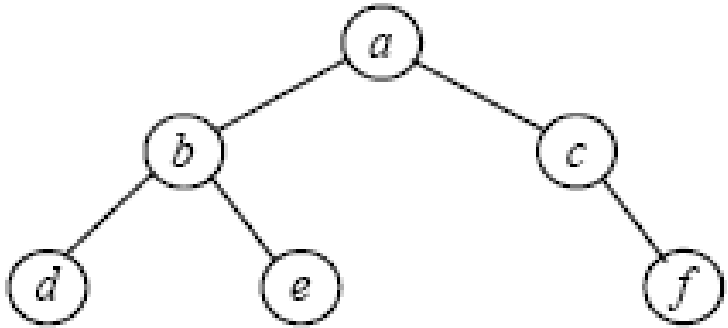
$Algorithm\ inorder(T)$
$if\ T \neq \emptyset$
$\quad \text{inorder}(T_{left})$
$\quad print(root\ of\ T)$
$\quad \text{inorder}(T_{right})$

$Algorithm\ Postorder(T)$
$if\ T \neq \emptyset$
$\quad Postorder(T_{left})$
$\quad Postrder(T_{right})$
$\quad print(root\ of\ T)$

Discussion

Traverse the following binary tree

a.. in preorder. b. in inorder. c. in postorder.



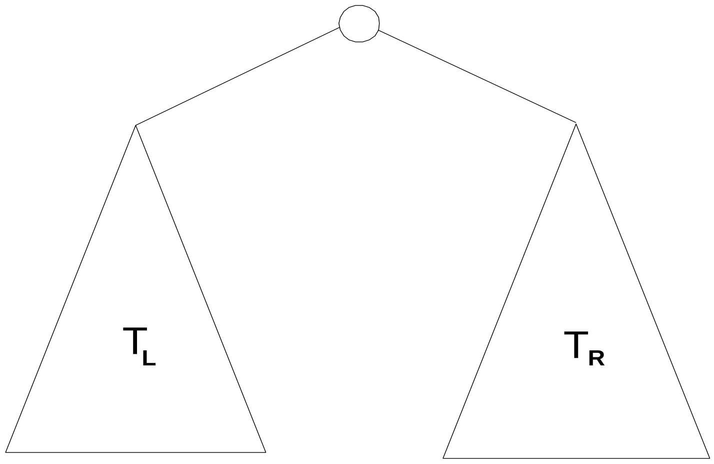
a. Preorder: $a \quad b \quad d \quad e \quad c \quad f$

b. Inorder: $d \quad b \quad e \quad a \quad c \quad f$

c. Postorder: $d \quad e \quad b \quad f \quad c \quad a$

Binary Tree Algorithms

Computing the height of a binary tree

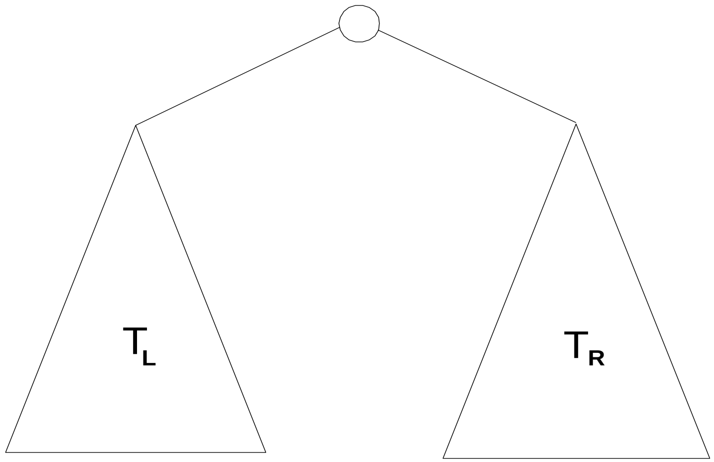


if $T = \emptyset \quad \rightarrow \quad h(\emptyset) = -1$

if $T \neq \emptyset \quad \rightarrow \quad h(T) = \max\{h(T_L),\ h(T_R)\} + 1$

Binary Tree Algorithms

Computing the height of a binary tree



if $T = \emptyset \quad \rightarrow \quad h(\emptyset) = -1$

if $T \neq \emptyset \quad \rightarrow \quad h(T) = \max\{h(T_L),\ h(T_R)\} + 1$

Efficiency: $\Theta(n)$

BST Search

Very efficient algorithm for searching *K* in *BST*:

If *K* = r.data, stop (successful search);

otherwise, continue searching by the same method

in left sub-tree if *K* < r.data

in right sub-tree if *K* > r.data

while t != $nill$ do

if K = t.data return t

else if K < t.data $t \leftarrow t.left$

else $t \leftarrow t.right$

return -1

Analysis of BST Search

- Average Time efficiency
  - $C(1) = 1$
  - $C(n) = 1 + C(\lfloor \frac{n}{2} \rfloor)$
  - $C(n) = \lceil \log_2(n+1) \rceil$
  - This is VERY fast: e.g., $C(10^6) = 20$

Discussion

Estimate how many times faster an average successful search will be in a sorted array of 100,000 elements if it is done by binary search versus sequential search.

Discussion

Estimate how many times faster an average successful search will be in a sorted array of 100,000 elements if it is done by binary search versus sequential search.

$$\frac{C_{avg}^{seq.}(n)}{C_{avg}^{bin.}(n)} \approx \frac{n/2}{\log_2 n} = (\text{for } n = 10^5)\frac{10^5/2}{\log_2 10^5} = \frac{1}{2*5}\frac{10^5}{\log_2 10} = \frac{10^4}{\log_2 10} \approx 3000.$$

Discussion

A version of the popular problem-solving task involves presenting people with an array of 42 pictures—seven rows of six pictures each—and asking them to identify the target picture by asking questions that can be answered yes or no. Further, people are then required to identify the picture with as few questions as possible. Suggest the most efficient algorithm for this problem and indicate the largest number of questions that may be necessary.

Discussion

A version of the popular problem-solving task involves presenting people with an array of 42 pictures—seven rows of six pictures each—and asking them to identify the target picture by asking questions that can be answered yes or no. Further, people are then required to identify the picture with as few questions as possible. Suggest the most efficient algorithm for this problem and indicate the largest number of questions that may be necessary.

Apply a two-way comparison version of binary search using the picture numbering. That is, assuming that pictures are numbered from 1 to 42, start with a question such as "Is the picture's number > 21?". The largest number of questions that may be required is 6. (Because the search can be assumed successful, one less comparison needs to be made than in *TwoWayBinarySearch*, yielding here $\lceil \log_2 42 \rceil = 6$.)

Multiplication of Large Integers

Consider the problem of multiplying two (large) n-digit integers represented by arrays of their digits such as:

A = 12345678901357986429   B = 87654321284820912836

Multiplication of Large Integers

Consider the problem of multiplying two (large) n-digit integers represented by arrays of their digits such as:

A = 12345678901357986429 B = 87654321284820912836

The grade-school algorithm:

$$
\begin{array}{rrrrrrr}
 & & & a_1 & a_2 \ldots & a_n \\
 & & & \underline{b_1} & \underline{b_2 \ldots} & \underline{b_n} \\
 & & (d_{10}) & d_{11} d_{12} \ldots & d_{1n} & \\
 & (d_{20}) & d_{21} d_{22} \ldots & d_{2n} & & \\
 & \ldots \ldots \ldots & \ldots \ldots \ldots \ldots & & & \\
\hline
(d_{n0}) & d_{n1} d_{n2} \ldots & d_{nn} & & &
\end{array}
$$

Multiplication of Large Integers

Consider the problem of multiplying two (large) n-digit integers represented by arrays of their digits such as:

A = 12345678901357986429 B = 87654321284820912836

The grade-school algorithm:

$$
\begin{array}{r}
a_1 \;\; a_2 \; \ldots \;\; a_n \\
\underline{b_1 \;\; b_2 \; \ldots \;\; b_n} \\
(d_{10})\; d_{11} d_{12} \ldots \; d_{1n} \\
(d_{20})\; d_{21} d_{22} \ldots \; d_{2n} \quad\quad \\
\ldots \; \ldots \; \ldots \; \ldots \; \ldots \; \ldots \; \ldots \quad\quad \\
\hline
(d_{n0})\; d_{n1} d_{n2} \ldots \; d_{nn} \quad\quad\quad\quad
\end{array}
$$

Efficiency: n^2 one-digit multiplications

First Divide-and-Conquer Algorithm

if $A = A_1 A_2$ and $B = B_1 B_2$

A and B are $n - digit$

A_1, A_2, B_1, B_2 are $\frac{n}{2}$-digit numbers

First Divide-and-Conquer Algorithm

if $A = A_1 A_2$ and $B = B_1 B_2$

A and B are $n - digit$

A_1, A_2, B_1, B_2 are $\frac{n}{2}$-digit numbers

$$A \times B = (A_1 \times B_1 \times 10^n) + (A_1 \times B_2 + A_2 \times B_1) \times 10^{\frac{n}{2}} + (A_2 \times B_2)$$

First Divide-and-Conquer Algorithm

if $A = A_1 A_2$ and $B = B_1 B_2$

A and B are $n - digit$

A_1, A_2, B_1, B_2 are $\frac{n}{2}$-digit numbers

$$A \times B = (A_1 \times B_1 \times 10^n) + (A_1 \times B_2 + A_2 \times B_1) \times 10^{\frac{n}{2}} + (A_2 \times B_2)$$

$A = 2135$, $B = 4014$

First Divide-and-Conquer Algorithm

if $A = A_1 A_2$ and $B = B_1 B_2$

A and B are $n - digit$

$A_1,\ A_2,\ B_1,\ B_2$ are $\frac{n}{2}$-digit numbers

$$A \times B = (A_1 \times B_1 \times 10^n) + (A_1 \times B_2 + A_2 \times B_1) \times 10^{\frac{n}{2}} + (A_2 \times B_2)$$

$A = 2135$, $B = 4014$

$A_1 = 21, A_2 = 35$, $B_1 = 40, B_2 = 14$

First Divide-and-Conquer Algorithm

if $A = A_1 A_2$ and $B = B_1 B_2$

A and B are $n - digit$

A_1, A_2, B_1, B_2 are $\frac{n}{2}$-digit numbers

$$A \times B = (A_1 \times B_1 \times 10^n) + (A_1 \times B_2 + A_2 \times B_1) \times 10^{\frac{n}{2}} + (A_2 \times B_2)$$

$A = 2135$, $B = 4014$

$A_1 = 21, A_2 = 35$, $B_1 = 40, B_2 = 14$

$A \times B =$

First Divide-and-Conquer Algorithm

if $A = A_1A_2$ and $B = B_1B_2$

A and B are $n - digit$

A_1, A_2, B_1, B_2 are $\frac{n}{2}$-digit numbers

$$A \times B = (A_1 \times B_1 \times 10^n) + (A_1 \times B_2 + A_2 \times B_1) \times 10^{\frac{n}{2}} + (A_2 \times B_2)$$

$A = 2135$, $B = 4014$

$A_1 = 21, A_2 = 35$, $B_1 = 40, B_2 = 14$

$A \times B = (21 \times 40 \times 10^4) +$

First Divide-and-Conquer Algorithm

if $A = A_1A_2$ and $B = B_1B_2$

A and B are $n - digit$

A_1, A_2, B_1, B_2 are $\frac{n}{2}$-digit numbers

$$A \times B = (A_1 \times B_1 \times 10^n) + (A_1 \times B_2 + A_2 \times B_1) \times 10^{\frac{n}{2}} + (A_2 \times B_2)$$

$A = 2135$, $B = 4014$

$A_1 = 21, A_2 = 35$, $B_1 = 40, B_2 = 14$

$A \times B = (21 \times 40 \times 10^4) + (21 \times 14 + 35 \times 40) \times 10^2 +$

First Divide-and-Conquer Algorithm

if $A = A_1A_2$ and $B = B_1B_2$

A and B are $n - digit$

A_1, A_2, B_1, B_2 are $\frac{n}{2}$-digit numbers

$$A \times B = (A_1 \times B_1 \times 10^n) + (A_1 \times B_2 + A_2 \times B_1) \times 10^{\frac{n}{2}} + (A_2 \times B_2)$$

$A = 2135$, $B = 4014$

$A_1 = 21, A_2 = 35$, $B_1 = 40, B_2 = 14$

$$A \times B = (21 \times 40 \times 10^4) + (21 \times 14 + 35 \times 40) \times 10^2 + (35 \times 14)$$

First Divide-and-Conquer Algorithm

if $A = A_1 A_2$ and $B = B_1 B_2$

A and B are $n - digit$

$A_1,\ A_2,\ B_1,\ B_2$ are $\frac{n}{2}$-digit numbers

$$A \times B = (A_1 \times B_1 \times 10^n) + (A_1 \times B_2 + A_2 \times B_1) \times 10^{\frac{n}{2}} + (A_2 \times B_2)$$

$A = 2135$, $B = 4014$

$A_1 = 21, A_2 = 35$, $B_1 = 40, B_2 = 14$

$A \times B = (21 \times 40 \times 10^4) + (21 \times 14 + 35 \times 40) \times 10^2 + (35 \times 14)$

$21 \times 40 =$

First Divide-and-Conquer Algorithm

if $A = A_1 A_2$ and $B = B_1 B_2$

A and B are $n - digit$

A_1, A_2, B_1, B_2 are $\frac{n}{2}$-digit numbers

$$A \times B = (A_1 \times B_1 \times 10^n) + (A_1 \times B_2 + A_2 \times B_1) \times 10^{\frac{n}{2}} + (A_2 \times B_2)$$

$A = 2135$, $B = 4014$

$A_1 = 21, A_2 = 35$, $B_1 = 40, B_2 = 14$

$$A \times B = (21 \times 40 \times 10^4) + (21 \times 14 + 35 \times 40) \times 10^2 + (35 \times 14)$$

$$21 \times 40 = \left(2 \times 4 \times 10^2\right) +$$

First Divide-and-Conquer Algorithm

if $A = A_1A_2$ and $B = B_1B_2$

A and B are $n - digit$

A_1, A_2, B_1, B_2 are $\frac{n}{2}$-digit numbers

$$A \times B = (A_1 \times B_1 \times 10^n) + (A_1 \times B_2 + A_2 \times B_1) \times 10^{\frac{n}{2}} + (A_2 \times B_2)$$

$A = 2135$, $B = 4014$

$A_1 = 21, A_2 = 35$, $B_1 = 40, B_2 = 14$

$A \times B = (21 \times 40 \times 10^4) + (21 \times 14 + 35 \times 40) \times 10^2 + (35 \times 14)$

$21 \times 40 = \left(2 \times 4 \times 10^2\right) + (2 \times 0 + 1 \times 4) \times 10 +$

First Divide-and-Conquer Algorithm

if $A = A_1 A_2$ and $B = B_1 B_2$

A and B are $n - digit$

A_1, A_2, B_1, B_2 are $\frac{n}{2}$-digit numbers

$$A \times B = (A_1 \times B_1 \times 10^n) + (A_1 \times B_2 + A_2 \times B_1) \times 10^{\frac{n}{2}} + (A_2 \times B_2)$$

$A = 2135$, $B = 4014$

$A_1 = 21, A_2 = 35$, $B_1 = 40, B_2 = 14$

$$A \times B = (21 \times 40 \times 10^4) + (21 \times 14 + 35 \times 40) \times 10^2 + (35 \times 14)$$

$$21 \times 40 = \left(2 \times 4 \times 10^2\right) + (2 \times 0 + 1 \times 4) \times 10 + (1 \times 0) =$$

First Divide-and-Conquer Algorithm

if $A = A_1 A_2$ and $B = B_1 B_2$

A and B are $n - digit$

A_1, A_2, B_1, B_2 are $\frac{n}{2}$-digit numbers

$$A \times B = (A_1 \times B_1 \times 10^n) + (A_1 \times B_2 + A_2 \times B_1) \times 10^{\frac{n}{2}} + (A_2 \times B_2)$$

$A = 2135$, $B = 4014$

$A_1 = 21, A_2 = 35$, $B_1 = 40, B_2 = 14$

$$A \times B = (21 \times 40 \times 10^4) + (21 \times 14 + 35 \times 40) \times 10^2 + (35 \times 14)$$

$$21 \times 40 = \left(2 \times 4 \times 10^2\right) + (2 \times 0 + 1 \times 4) \times 10 + (1 \times 0) = 800 + 40 + 0 = 840$$

First Divide-and-Conquer Algorithm

if $A = A_1A_2$ and $B = B_1B_2$

A and B are $n - digit$

A_1, A_2, B_1, B_2 are $\frac{n}{2}$-digit numbers

$$A \times B = (A_1 \times B_1 \times 10^n) + (A_1 \times B_2 + A_2 \times B_1) \times 10^{\frac{n}{2}} + (A_2 \times B_2)$$

$A = 2135$, $B = 4014$

$A_1 = 21, A_2 = 35$, $B_1 = 40, B_2 = 14$

$A \times B = (21 \times 40 \times 10^4) + (21 \times 14 + 35 \times 40) \times 10^2 + (35 \times 14)$

$21 \times 40 = \left(2 \times 4 \times 10^2\right) + (2 \times 0 + 1 \times 4) \times 10 + (1 \times 0) = 800 + 40 + 0 = 840$

$21 \times 14 =$

First Divide-and-Conquer Algorithm

if $A = A_1A_2$ and $B = B_1B_2$

A and B are $n - digit$

A_1, A_2, B_1, B_2 are $\frac{n}{2}$-digit numbers

$$A \times B = (A_1 \times B_1 \times 10^n) + (A_1 \times B_2 + A_2 \times B_1) \times 10^{\frac{n}{2}} + (A_2 \times B_2)$$

$A = 2135$, $B = 4014$

$A_1 = 21, A_2 = 35$, $B_1 = 40, B_2 = 14$

$A \times B = (21 \times 40 \times 10^4) + (21 \times 14 + 35 \times 40) \times 10^2 + (35 \times 14)$

$21 \times 40 = \left(2 \times 4 \times 10^2\right) + (2 \times 0 + 1 \times 4) \times 10 + (1 \times 0) = 800 + 40 + 0 = 840$

$21 \times 14 = \left(2 \times 1 \times 10^2\right) +$

First Divide-and-Conquer Algorithm

if $A = A_1A_2$ and $B = B_1B_2$

A and B are $n - digit$

A_1, A_2, B_1, B_2 are $\frac{n}{2}$-digit numbers

$$A \times B = (A_1 \times B_1 \times 10^n) + (A_1 \times B_2 + A_2 \times B_1) \times 10^{\frac{n}{2}} + (A_2 \times B_2)$$

$A = 2135$, $B = 4014$

$A_1 = 21, A_2 = 35$, $B_1 = 40, B_2 = 14$

$$A \times B = (21 \times 40 \times 10^4) + (21 \times 14 + 35 \times 40) \times 10^2 + (35 \times 14)$$

$$21 \times 40 = \left(2 \times 4 \times 10^2\right) + (2 \times 0 + 1 \times 4) \times 10 + (1 \times 0) = 800 + 40 + 0 = 840$$

$$21 \times 14 = \left(2 \times 1 \times 10^2\right) + (2 \times 4 + 1 \times 1) \times 10 +$$

First Divide-and-Conquer Algorithm

if $A = A_1 A_2$ and $B = B_1 B_2$

A and B are $n - digit$

A_1, A_2, B_1, B_2 are $\frac{n}{2}$-digit numbers

$$A \times B = (A_1 \times B_1 \times 10^n) + (A_1 \times B_2 + A_2 \times B_1) \times 10^{\frac{n}{2}} + (A_2 \times B_2)$$

$A = 2135$, $B = 4014$

$A_1 = 21, A_2 = 35$, $B_1 = 40, B_2 = 14$

$$A \times B = (21 \times 40 \times 10^4) + (21 \times 14 + 35 \times 40) \times 10^2 + (35 \times 14)$$

$$21 \times 40 = \left(2 \times 4 \times 10^2\right) + (2 \times 0 + 1 \times 4) \times 10 + (1 \times 0) = 800 + 40 + 0 = 840$$

$$21 \times 14 = \left(2 \times 1 \times 10^2\right) + (2 \times 4 + 1 \times 1) \times 10 + (1 \times 4) =$$

First Divide-and-Conquer Algorithm

if $A = A_1A_2$ and $B = B_1B_2$

A and B are $n - digit$

A_1, A_2, B_1, B_2 are $\frac{n}{2}$-digit numbers

$$A \times B = (A_1 \times B_1 \times 10^n) + (A_1 \times B_2 + A_2 \times B_1) \times 10^{\frac{n}{2}} + (A_2 \times B_2)$$

$A = 2135$, $B = 4014$

$A_1 = 21, A_2 = 35$, $B_1 = 40, B_2 = 14$

$$A \times B = (21 \times 40 \times 10^4) + (21 \times 14 + 35 \times 40) \times 10^2 + (35 \times 14)$$

$$21 \times 40 = \left(2 \times 4 \times 10^2\right) + (2 \times 0 + 1 \times 4) \times 10 + (1 \times 0) = 800 + 40 + 0 = 840$$

$$21 \times 14 = \left(2 \times 1 \times 10^2\right) + (2 \times 4 + 1 \times 1) \times 10 + (1 \times 4) = 200 + 90 + 4 = 294$$

First Divide-and-Conquer Algorithm

if $A = A_1A_2$ and $B = B_1B_2$

A and B are $n - digit$

A_1, A_2, B_1, B_2 are $\frac{n}{2}$-digit numbers

$$A \times B = (A_1 \times B_1 \times 10^n) + (A_1 \times B_2 + A_2 \times B_1) \times 10^{\frac{n}{2}} + (A_2 \times B_2)$$

$A = 2135$, $B = 4014$

$A_1 = 21, A_2 = 35$, $B_1 = 40, B_2 = 14$

$$A \times B = (21 \times 40 \times 10^4) + (21 \times 14 + 35 \times 40) \times 10^2 + (35 \times 14)$$

$$21 \times 40 = \left(2 \times 4 \times 10^2\right) + (2 \times 0 + 1 \times 4) \times 10 + (1 \times 0) = 800 + 40 + 0 = 840$$

$$21 \times 14 = \left(2 \times 1 \times 10^2\right) + (2 \times 4 + 1 \times 1) \times 10 + (1 \times 4) = 200 + 90 + 4 = 294$$

$$35 \times 40 =$$

First Divide-and-Conquer Algorithm

if $A = A_1A_2$ and $B = B_1B_2$

A and B are $n - digit$

$A_1,\ A_2,\ B_1,\ B_2$ are $\frac{n}{2}$-digit numbers

$$A \times B = (A_1 \times B_1 \times 10^n) + (A_1 \times B_2 + A_2 \times B_1) \times 10^{\frac{n}{2}} + (A_2 \times B_2)$$

$A = 2135$, $B = 4014$

$A_1 = 21, A_2 = 35$, $B_1 = 40, B_2 = 14$

$$A \times B = (21 \times 40 \times 10^4) + (21 \times 14 + 35 \times 40) \times 10^2 + (35 \times 14)$$

$$21 \times 40 = \left(2 \times 4 \times 10^2\right) + (2 \times 0 + 1 \times 4) \times 10 + (1 \times 0) = 800 + 40 + 0 = 840$$
$$21 \times 14 = \left(2 \times 1 \times 10^2\right) + (2 \times 4 + 1 \times 1) \times 10 + (1 \times 4) = 200 + 90 + 4 = 294$$
$$35 \times 40 = \left(3 \times 4 \times 10^2\right) + (3 \times 0 + 5 \times 4) \times 10 + (5 \times 0) =$$

First Divide-and-Conquer Algorithm

if $A = A_1A_2$ and $B = B_1B_2$

A and B are $n - digit$

A_1, A_2, B_1, B_2 are $\frac{n}{2}$-digit numbers

$$A \times B = (A_1 \times B_1 \times 10^n) + (A_1 \times B_2 + A_2 \times B_1) \times 10^{\frac{n}{2}} + (A_2 \times B_2)$$

$A = 2135$, $B = 4014$

$A_1 = 21, A_2 = 35$, $B_1 = 40, B_2 = 14$

$$A \times B = (21 \times 40 \times 10^4) + (21 \times 14 + 35 \times 40) \times 10^2 + (35 \times 14)$$

$$21 \times 40 = \left(2 \times 4 \times 10^2\right) + (2 \times 0 + 1 \times 4) \times 10 + (1 \times 0) = 800 + 40 + 0 = 840$$

$$21 \times 14 = \left(2 \times 1 \times 10^2\right) + (2 \times 4 + 1 \times 1) \times 10 + (1 \times 4) = 200 + 90 + 4 = 294$$

$$35 \times 40 = \left(3 \times 4 \times 10^2\right) + (3 \times 0 + 5 \times 4) \times 10 + (5 \times 0) = 1200 + 200 + 0 = 1400$$

First Divide-and-Conquer Algorithm

if $A = A_1 A_2$ and $B = B_1 B_2$

A and B are $n - digit$

A_1, A_2, B_1, B_2 are $\frac{n}{2}$-digit numbers

$$A \times B = (A_1 \times B_1 \times 10^n) + (A_1 \times B_2 + A_2 \times B_1) \times 10^{\frac{n}{2}} + (A_2 \times B_2)$$

$A = 2135$, $B = 4014$

$A_1 = 21, A_2 = 35$, $B_1 = 40, B_2 = 14$

$A \times B = (21 \times 40 \times 10^4) + (21 \times 14 + 35 \times 40) \times 10^2 + (35 \times 14)$

$21 \times 40 = (2 \times 4 \times 10^2) + (2 \times 0 + 1 \times 4) \times 10 + (1 \times 0) = 800 + 40 + 0 = 840$

$21 \times 14 = (2 \times 1 \times 10^2) + (2 \times 4 + 1 \times 1) \times 10 + (1 \times 4) = 200 + 90 + 4 = 294$

$35 \times 40 = (3 \times 4 \times 10^2) + (3 \times 0 + 5 \times 4) \times 10 + (5 \times 0) = 1200 + 200 + 0 = 1400$

$35 \times 14 =$

First Divide-and-Conquer Algorithm

if $A = A_1A_2$ and $B = B_1B_2$

A and B are $n - digit$

A_1, A_2, B_1, B_2 are $\frac{n}{2}$-digit numbers

$$A \times B = (A_1 \times B_1 \times 10^n) + (A_1 \times B_2 + A_2 \times B_1) \times 10^{\frac{n}{2}} + (A_2 \times B_2)$$

$A = 2135$, $B = 4014$

$A_1 = 21, A_2 = 35$, $B_1 = 40, B_2 = 14$

$$A \times B = (21 \times 40 \times 10^4) + (21 \times 14 + 35 \times 40) \times 10^2 + (35 \times 14)$$

$$21 \times 40 = (2 \times 4 \times 10^2) + (2 \times 0 + 1 \times 4) \times 10 + (1 \times 0) = 800 + 40 + 0 = 840$$

$$21 \times 14 = (2 \times 1 \times 10^2) + (2 \times 4 + 1 \times 1) \times 10 + (1 \times 4) = 200 + 90 + 4 = 294$$

$$35 \times 40 = (3 \times 4 \times 10^2) + (3 \times 0 + 5 \times 4) \times 10 + (5 \times 0) = 1200 + 200 + 0 = 1400$$

$$35 \times 14 = (3 \times 1 \times 10^2) + (3 \times 4 + 5 \times 1) \times 10 + (5 \times 4) =$$

First Divide-and-Conquer Algorithm

if $A = A_1 A_2$ and $B = B_1 B_2$

A and B are $n - digit$

A_1, A_2, B_1, B_2 are $\frac{n}{2}$-digit numbers

$$A \times B = (A_1 \times B_1 \times 10^n) + (A_1 \times B_2 + A_2 \times B_1) \times 10^{\frac{n}{2}} + (A_2 \times B_2)$$

$A = 2135$, $B = 4014$

$A_1 = 21, A_2 = 35$, $B_1 = 40, B_2 = 14$

$$A \times B = (21 \times 40 \times 10^4) + (21 \times 14 + 35 \times 40) \times 10^2 + (35 \times 14)$$

$$21 \times 40 = (2 \times 4 \times 10^2) + (2 \times 0 + 1 \times 4) \times 10 + (1 \times 0) = 800 + 40 + 0 = 840$$
$$21 \times 14 = (2 \times 1 \times 10^2) + (2 \times 4 + 1 \times 1) \times 10 + (1 \times 4) = 200 + 90 + 4 = 294$$
$$35 \times 40 = (3 \times 4 \times 10^2) + (3 \times 0 + 5 \times 4) \times 10 + (5 \times 0) = 1200 + 200 + 0 = 1400$$
$$35 \times 14 = (3 \times 1 \times 10^2) + (3 \times 4 + 5 \times 1) \times 10 + (5 \times 4) = 300 + 170 + 20 = 490$$

First Divide-and-Conquer Algorithm

if $A = A_1A_2$ and $B = B_1B_2$

A and B are $n - digit$

A_1, A_2, B_1, B_2 are $\frac{n}{2}$-digit numbers

$$A \times B = (A_1 \times B_1 \times 10^n) + (A_1 \times B_2 + A_2 \times B_1) \times 10^{\frac{n}{2}} + (A_2 \times B_2)$$

$A = 2135$, $B = 4014$

$A_1 = 21, A_2 = 35$, $B_1 = 40, B_2 = 14$

$$A \times B = (21 \times 40 \times 10^4) + (21 \times 14 + 35 \times 40) \times 10^2 + (35 \times 14)$$

$$21 \times 40 = \left(2 \times 4 \times 10^2\right) + (2 \times 0 + 1 \times 4) \times 10 + (1 \times 0) = 800 + 40 + 0 = 840$$

$$21 \times 14 = \left(2 \times 1 \times 10^2\right) + (2 \times 4 + 1 \times 1) \times 10 + (1 \times 4) = 200 + 90 + 4 = 294$$

$$35 \times 40 = \left(3 \times 4 \times 10^2\right) + (3 \times 0 + 5 \times 4) \times 10 + (5 \times 0) = 1200 + 200 + 0 = 1400$$

$$35 \times 14 = \left(3 \times 1 \times 10^2\right) + (3 \times 4 + 5 \times 1) \times 10 + (5 \times 4) = 300 + 170 + 20 = 490$$

$$A \times B = (840 \times 10^4) + (294 + 1400) \times 10^2 + (490) =$$

First Divide-and-Conquer Algorithm

if $A = A_1A_2$ and $B = B_1B_2$

A and B are $n - digit$

A_1, A_2, B_1, B_2 are $\frac{n}{2}$-digit numbers

$$A \times B = (A_1 \times B_1 \times 10^n) + (A_1 \times B_2 + A_2 \times B_1) \times 10^{\frac{n}{2}} + (A_2 \times B_2)$$

$A = 2135$, $B = 4014$

$A_1 = 21, A_2 = 35$, $B_1 = 40, B_2 = 14$

$$A \times B = (21 \times 40 \times 10^4) + (21 \times 14 + 35 \times 40) \times 10^2 + (35 \times 14)$$

$$21 \times 40 = \left(2 \times 4 \times 10^2\right) + (2 \times 0 + 1 \times 4) \times 10 + (1 \times 0) = 800 + 40 + 0 = 840$$

$$21 \times 14 = \left(2 \times 1 \times 10^2\right) + (2 \times 4 + 1 \times 1) \times 10 + (1 \times 4) = 200 + 90 + 4 = 294$$

$$35 \times 40 = \left(3 \times 4 \times 10^2\right) + (3 \times 0 + 5 \times 4) \times 10 + (5 \times 0) = 1200 + 200 + 0 = 1400$$

$$35 \times 14 = \left(3 \times 1 \times 10^2\right) + (3 \times 4 + 5 \times 1) \times 10 + (5 \times 4) = 300 + 170 + 20 = 490$$

$$A \times B = (840 \times 10^4) + (294 + 1400) \times 10^2 + (490) = 8569890$$

First Divide-and-Conquer Algorithm

if $A = A_1A_2$ and $B = B_1B_2$

A and B are $n - digit$

A_1, A_2, B_1, B_2 are $\frac{n}{2}$-digit numbers

$$A \times B = (A_1 \times B_1 \times 10^n) + (A_1 \times B_2 + A_2 \times B_1) \times 10^{\frac{n}{2}} + (A_2 \times B_2)$$

First Divide-and-Conquer Algorithm

if $A = A_1A_2$ and $B = B_1B_2$

A and B are $n - digit$

A_1, A_2, B_1, B_2 are $\frac{n}{2}$-digit numbers

$$A \times B = (A_1 \times B_1 \times 10^n) + (A_1 \times B_2 + A_2 \times B_1) \times 10^{\frac{n}{2}} + (A_2 \times B_2)$$

Recurrence for the number of one-digit multiplications M(n):

$M(1) = 1$

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

$M(n) = n^2$

Second Divide-and-Conquer Algorithm

The idea is to decrease the number of multiplications from 4 to 3:

$$(A_1 + A_2) \times (B_1 + B_2) = A_1 \times B_1 + (A_1 \times B_2 + A_2 \times B_1) + A_2 \times B_2$$

$$(A_1 \times B_2 + A_2 \times B_1) = (A_1 + A_2) \times (B_1 + B_2) - A_1 \times B_1 - A_2 \times B_2$$

which requires only 3 multiplications at the expense of three extra add/sub.

$$A \times B = A_1 \times B_1 \times 10^n + \big((A_1 + A_2) \times (B_1 + B_2) - A_1 \times B_1 - A_2 \times B_2\big) \times 10^{\frac{n}{2}} + A_2 \times B_2$$

Second Divide-and-Conquer Algorithm

The idea is to decrease the number of multiplications from 4 to 3:

$$(A_1 + A_2) \times (B_1 + B_2) = A_1 \times B_1 + (A_1 \times B_2 + A_2 \times B_1) + A_2 \times B_2$$

$$(A_1 \times B_2 + A_2 \times B_1) = (A_1 + A_2) \times (B_1 + B_2) - A_1 \times B_1 - A_2 \times B_2$$

which requires only 3 multiplications at the expense of three extra add/sub.

$$A \times B = A_1 \times B_1 \times 10^n + \left((A_1 + A_2) \times (B_1 + B_2) - A_1 \times B_1 - A_2 \times B_2\right) \times 10^{\frac{n}{2}} + A_2 \times B_2$$

Recurrence for the number of multiplications M(n):

$M(1) = 1$

$M(n) = 3M\left(\frac{n}{2}\right)+1$

$M(n) = 3^{\log_2 n} = n^{\log_2 3} \approx n^{1.585}$

Discussion

Compute $2101 * 1130$ by applying the divide-and-conquer algorithm outlined in the text.

$$A \times B = A_1 \times B_1 \times 10^n + \left((A_1 + A_2) \times (B_1 + B_2) - A_1 \times B_1 - A_2 \times B_2\right) \times 10^{\frac{n}{2}} + A_2 \times B_2$$

Discussion

Compute $2101 * 1130$ by applying the divide-and-conquer algorithm outlined in the text.

For $2101 * 1130$:

$$\begin{aligned} c_2 &= 21 * 11 \\ c_0 &= 01 * 30 \\ c_1 &= (21 + 01) * (11 + 30) - (c_2 + c_0) = 22 * 41 - 21 * 11 - 01 * 30. \end{aligned}$$

For $21 * 11$:

$$\begin{aligned} c_2 &= 2 * 1 = 2 \\ c_0 &= 1 * 1 = 1 \\ c_1 &= (2 + 1) * (1 + 1) - (2 + 1) = 3 * 2 - 3 = 3. \\ \text{So, } 21 * 11 &= 2 \cdot 10^2 + 3 \cdot 10^1 + 1 = 231. \end{aligned}$$

For $01 * 30$:

$$\begin{aligned} c_2 &= 0 * 3 = 0 \\ c_0 &= 1 * 0 = 0 \\ c_1 &= (0 + 1) * (3 + 0) - (0 + 0) = 1 * 3 - 0 = 3. \\ \text{So, } 01 * 30 &= 0 \cdot 10^2 + 3 \cdot 10^1 + 0 = 30. \end{aligned}$$

For $22 * 41$:

$$\begin{aligned} c_2 &= 2 * 4 = 8 \\ c_0 &= 2 * 1 = 2 \\ c_1 &= (2 + 2) * (4 + 1) - (8 + 2) = 4 * 5 - 10 = 10. \\ \text{So, } 22 * 41 &= 8 \cdot 10^2 + 10 \cdot 10^1 + 2 = 902. \end{aligned}$$

Hence

$$2101 * 1130 = 231 \cdot 10^4 + (902 - 231 - 30) \cdot 10^2 + 30 = 2,374,130.$$