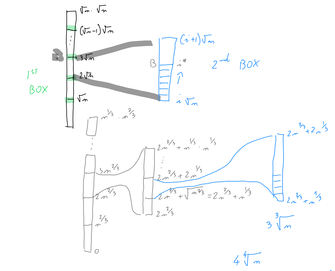


$4\sqrt[4]{n}$

$\boxed{k\sqrt[k]{n}}$

$\Downarrow$

$\lg n \cdot \sqrt[\lg n]{n}$

$\lg n \cdot O(1) = O(\lg n)$

$k = \lg n$

$n = 2^{\lg n}$

$(n)^{\frac{1}{\lg n}} = 2^{\lg n \cdot \frac{1}{\lg n}}$

$= 2$

GRADE SCHOOL

BASE 10

$$\begin{array}{r} 163 + \\ 453 \\ \hline 616 \end{array}$$

BASE 2

$$\begin{array}{r} 1011 + \\ 0101 \\ \hline 10000 \end{array}$$

BASE 2

$$\begin{array}{r} 1100 \cdot \\ 0111 \\ \hline 1100 \\ 1100 \\ 1100 \\ 0000 \\ \hline 1010100 \end{array}$$

$O(n^2)$

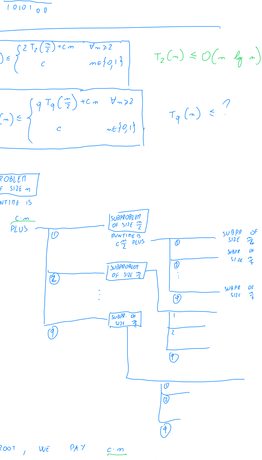
$\boxed{R_2}$ $T_2(n) \le \begin{cases} 2T_2(\frac{n}{2}) + cn & \forall n \ge 2 \\ c & n \in \{0,1\} \end{cases}$ $\qquad T_2(n) \le O(n \lg n)$

$\boxed{R_q}$ $q \ge 2$ $\quad T_q(n) \le \begin{cases} qT_q(\frac{n}{q}) + cn & \forall n \ge 2 \\ c & n \in \{0,1\} \end{cases}$ $\qquad T_q(n) \le ?$

PROBLEM OF SIZE n

RUNTIME IS

cn PLUS



AT THE ROOT, WE PAY cn

AT THE FIRST LEVEL, WE HAVE q NODES, EACH OF WHICH PAYS $c\frac{n}{2}$

" " 2nd " , " " q^2 " , " " " " $c\frac{n}{4}$

⋮

THE TOTAL COST AT LEVEL $i \in \{0,1,2,\dots\}$ IS THEN

$$L_i = c\frac{n}{2^i} q^i$$

SUPPOSE THAT $q \ge 3$. THEN, THE TOTAL RUNTIME IS

$$\begin{aligned}
\sum_{i=0}^{\lceil \lg n \rceil} L_i &= \sum_{i=0}^{\lceil \lg n \rceil} \left(c\frac{n}{2^i} q^i\right) \\
&= cn \sum_{i=0}^{\lceil \lg n \rceil} \left(\frac{q}{2}\right)^i \\
&= cn \frac{\left(\frac{q}{2}\right)^{\lceil \lg n \rceil + 1} - 1}{\frac{q}{2} - 1} \\
&= 2cn \frac{\left(\frac{q}{2}\right)^{\lceil \lg n \rceil + 1} - 1}{q - 2} \\
&\le 2cn \frac{\left(\frac{q}{2}\right)^{\lg n + 2}}{q-2} \\
&= \frac{2c\left(\frac{q}{2}\right)^2}{q-2} \cdot n \cdot \left(\frac{q}{2}\right)^{\lg n} \\
&= \frac{2c\left(\frac{q}{2}\right)^2}{q-2} \cdot n \cdot \left(2^{\lg n \cdot \lg \frac{q}{2}}\right) \\
&= \frac{2c\left(\frac{q}{2}\right)^2}{q-2} \cdot n \cdot \left(2^{\lg n}\right)^{\lg \frac{q}{2}} \\
&= \frac{2c\left(\frac{q}{2}\right)^2}{q-2} \cdot n \cdot n^{\lg \frac{q}{2}} = \\
&= \frac{2c\left(\frac{q}{2}\right)^2}{q-2} \cdot n \cdot n^{\lg q - \lg 2} \\
&= \frac{2c\left(\frac{q}{2}\right)^2}{q-2} \cdot n \cdot n^{\lg q - 1} = O(n^{\lg q})
\end{aligned}$$

$$\sum_{i=0}^{k} \alpha^i = \frac{\alpha^{k+1} - 1}{\alpha - 1} \quad \forall \alpha > 1$$

$$(x^a)^b = x^{ab} = (x^b)^a$$

THM: $R_q(n) \le O(n^{\lg q})$ $\quad \forall q \ge 3$

WE WANT TO MULTIPLY TWO INTEGERS x, y OF n BITS EACH.

LET $M = x \cdot y$.

WE APPLY A DIVIDE-ET-IMPERA APPROACH BY SPLITTING x AND y IN TWO HALVES EACH.

$x = 100\,110$ $\qquad x_1 = 100$ $\qquad x_0 = 110$

$y = 011\,010$ $\qquad y_1 = 011$ $\qquad y_0 = 010$

THEN, $x = x_1 \cdot 2^{\frac{n}{2}} + x_0$ AND $y = y_1 \cdot 2^{\frac{n}{2}} + y_0$. THUS,

$$\begin{aligned}
x \cdot y &= \left(x_1 \cdot 2^{\frac{n}{2}} + x_0\right)\left(y_1 \cdot 2^{\frac{n}{2}} + y_0\right) \\
&= x_1 y_1 \cdot 2^n + x_1 y_0\, 2^{\frac{n}{2}} + x_0 y_1 \cdot 2^{\frac{n}{2}} + x_0 y_0
\end{aligned}$$

WE REDUCED THE PROBLEM OF MULTIPLYING TWO n-DIGITS NUMBERS INTO

(i) THE COMPUTATION OF 4 PRODUCTS OF PAIRS OF $\frac{n}{2}$-DIGITS BINARY NUMBERS $(x_1 \cdot y_1,\ x_1 \cdot y_0,\ x_0 y_1,\ x_0 y_0)$

(ii) 3 MULTIPLICATIONS BY POWERS OF 2;

(iii) 3 ADDITIONS.

NOW, (ii) AND (iii) CAN BE COMPUTED IN $O(n)$ TIME.

WHAT ABOUT (i)?

$$T_4(n) \le 4\, T_4\left(\frac{n}{2}\right) + O(n)$$

AS WE PROVED, $T_4(n) \le O(n^{\lg 4}) = O(n^2)$.

WE WANT TO HAVE FEWER SUBPROBLEMS!

$$M = x_1 y_1\, 2^n + (x_1 y_0 + x_0 y_1)\, 2^{\frac{n}{2}} + x_0 y_0 = x \cdot y$$

$$M_2 = (x_1 + x_0)(y_1 + y_0) = x_1 y_1 + x_1 y_0 + y_1 x_0 + x_0 y_0$$

$$M_1 = x_1 y_1$$

$$M_0 = x_0 y_0$$

$$M_2 - M_1 - M_0 = x_1 y_0 + y_1 x_0$$

THUS,

$$M = x \cdot y = M_1\, 2^n + (M_2 - M_1 - M_0)\, 2^{\frac{n}{2}} + M_0$$

THE RUNTIME IS NOW $T_3(n) \le O(n^{\lg 3}) = O(n^{1.58})$

RECURSIVE-MULTIPLY (x, y):

LET $x = x_1 \cdot 2^{\frac{n}{2}} + x_0$ AND $y = y_1 \cdot 2^{\frac{n}{2}} + y_0$	$O(n)$
COMPUTE $S_x = x_1 + x_0$ AND $S_y = y_1 + y_0$	$O(n)$
M_2 : RECURSIVE-MULTIPLY (S_x, S_y)	$T_3(\frac{n}{2})$
M_1 : RECURSIVE-MULTIPLY (x_1, y_1)	$T_3(\frac{n}{2})$
M_0 : RECURSIVE-MULTIPLY (x_0, y_0)	$T_3(\frac{n}{2})$
RETURN $M_1 \cdot 2^n + (M_2 - M_1 - M_0)\, 2^{\frac{n}{2}} + M_0$	$O(n)$

$$T_3(n) \le 3\, T_3\left(\frac{n}{2}\right) + O(n) \le O(n^{\lg 3})$$

THE BEST KNOWN ALGORITHM FOR INTEGER MULTIPLICATION TAKES TIME $n \lg n \cdot 2^{O(\lg^* n)}$

$\lg^* n$ IS THE MINIMUM INTEGER i SUCH THAT $\underbrace{\lg(\lg(\cdots(\lg(n))\cdots))}_{i} < 2$

$$\lg^*\left(2^{2^{2^{2^{2}}}}\right) \approx 5$$