$\boxed{\text{19 Feb 2024}}$ Karatsuba's Multiplication Algorithm

<u>Announcements</u> Prelim 1 tomorrow (Tues) 7:30–9:00

URIS G01 — all 5820 students + 4820 A–L

OLIN 155 — 4820 M–Q

OLIN 165 — 4820 R–Z

For urgent course administration matters, email Prof Kim, Prof Kleinberg, Sara Perkins

mpk76 — rdk2 — sep247

Homework 3 solution set to be posted on Canvas immediately after today's lecture.

$$
\begin{array}{r}
11 \\
\times\ 13 \\
\hline
33 \\
11 \\
\hline
143
\end{array}
\qquad\qquad
\left.
\begin{array}{r}
1011 \\
1101 \\
\hline
1011 \\
0000 \\
1011 \\
1011 \\
\hline
10001111
\end{array}
\right\} \; O(n^2)
$$

"2" "3"

$$\underbrace{10}\underbrace{11}$$

$$11 = 2 \cdot 4 + 3 \cdot 1$$

Binary number $D = \underbrace{d_n \cdots d_{\frac{n}{2}+1}}_{D_1} \underbrace{d_{n/2} \cdots d_2 d_1 d_0}_{D_0}$

$$D = D_1 \cdot 2^{n/2} + D_0 \cdot 1.$$

To multiply $A * B$, write

$$A = A_1 \cdot 2^{n/2} + A_0$$

$$B = B_1 \cdot 2^{n/2} + B_0$$

$$AB = A_1 B_1 \cdot 2^n + A_0 B_1 2^{n/2} + A_1 B_0 \cdot 2^{n/2} + A_0 B_0$$

$$= A_1 B_1 \cdot 2^n + (A_0 B_1 + A_1 B_0) 2^{n/2} + A_0 B_0$$

$$T(n) = 4T\left(\tfrac{n}{2}\right) + c \cdot n$$

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

Let

$$P_0 = A_0 B_0$$

$$P_1 = A_1 B_1$$

$$P_2 = (A_1 + A_0)(B_1 + B_0)$$

$$= A_1 B_1 + (A_0 B_1 + A_1 B_0) + A_0 B_0$$

$$A_0 B_1 + A_1 B_0 = P_2 - P_1 - P_0$$

Karatsuba's algorithm

Given n-digit binary numbers A, B

0. Base case: 1-digit numbers.
1. Write $A = A_1 \cdot 2^{n/2} + A_0$, $B = B_1 \cdot 2^{n/2} + B_0$.

   where A_0, A_1, B_0, B_1 have $\le \frac{n}{2}$ digits.
2. Recursively, using Karatsuba, multiply

$$A_0 \cdot B_0 = P_0$$

$$A_1 \cdot B_1 = P_1$$

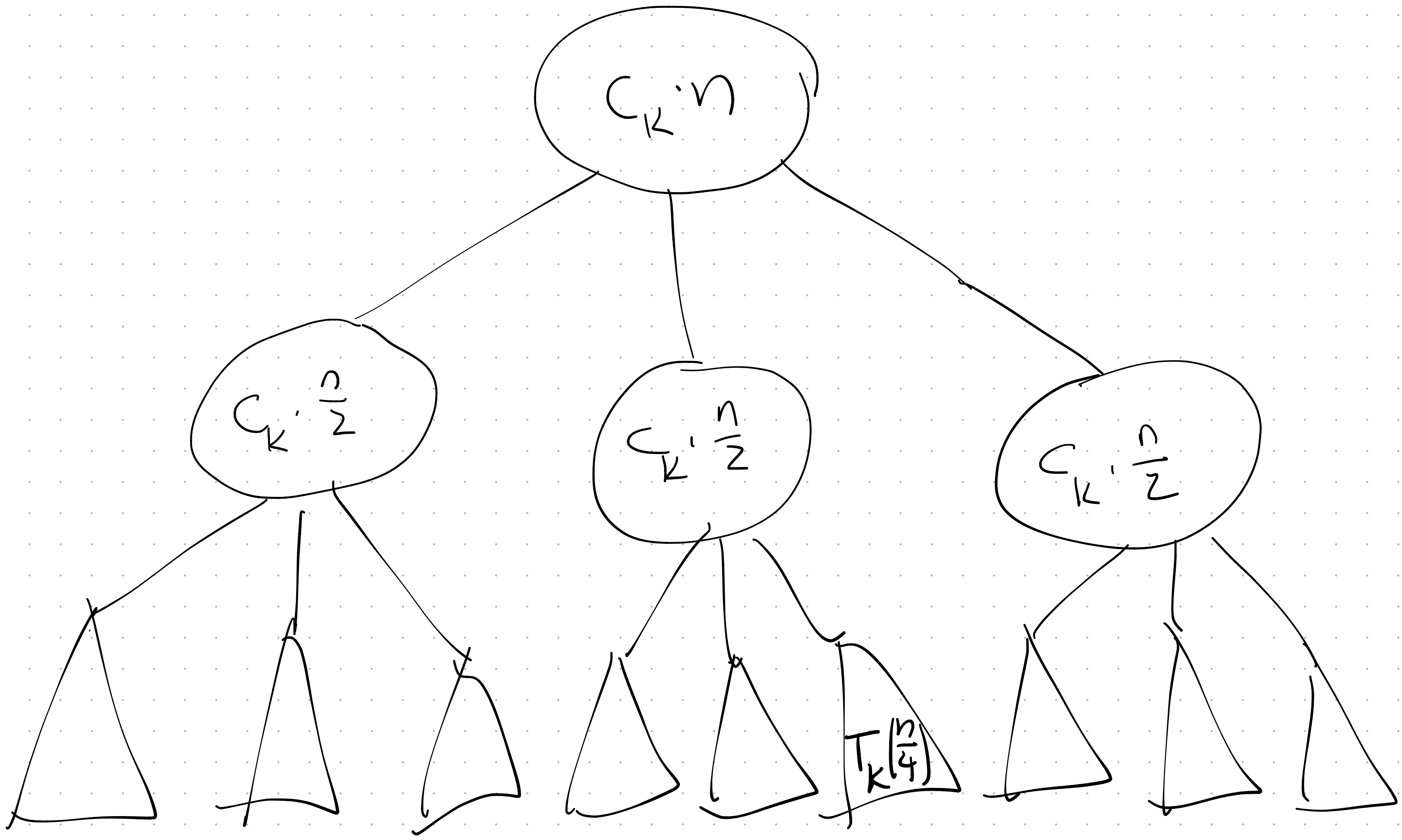
$$(A_0 + A_1)(B_0 + B_1) = P_2$$

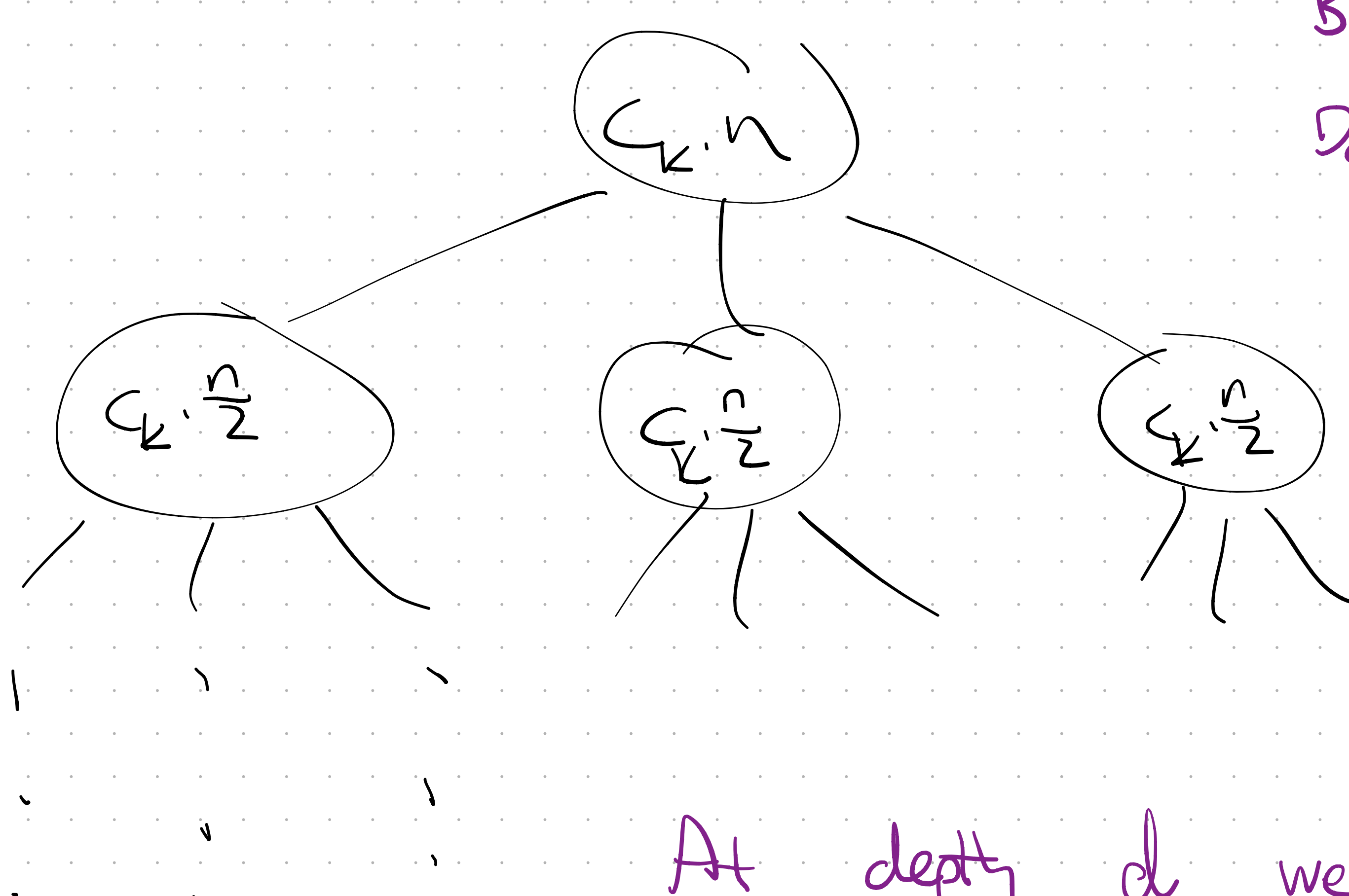
3. Compute $AB = P_1 \cdot 2^n + (P_2 - P_1 - P_0) \cdot 2^{\frac{n}{2}} + P_0$

Running time recurrence...

$$T_K(n) = \underbrace{2 \cdot T_K\left(\frac{n}{2}\right)}_{P_0 \,\&\, P_1} + \underbrace{T_K\left(\frac{n}{2} + 1\right)}_{P_2} + c_K n$$

$$\text{"="} \quad 3 \cdot T_K\left(\frac{n}{2}\right) + c_K \cdot n.$$





Branching factor 3
Depth $\log_2(n)$.

At depth d we have 3^d tree nodes, each representing $c_k \cdot \frac{n}{2^d}$ computational steps.

$$T_k(n) \leq \sum_{d=0}^{\log_2(n)} 3^d \cdot c_k \cdot \frac{n}{2^d} = c_k \cdot n \sum_{d=0}^{\log_2(n)} \left(\frac{3}{2}\right)^d$$

$$= c_K \cdot n \cdot \frac{(3/2)^{\log_2(n)+1} - 1}{3/2 - 1}$$

$$= O\left(\cancel{c_K \cdot n} \cdot \left(\cancel{\frac{3}{2}}\right)^{\log_2(n)}\right) \qquad O(n^{1.59})$$

$$= O\left(3^{\log_2(n)}\right) = O\left(n^{\log_2(3)}\right)$$

$$O\left(\left(2^{\log_2(3)}\right)^{\log_2(n)}\right) = O\left(\left(2^{\log_2(n)}\right)^{\log_2(3)}\right)$$

$$\left(\frac{3}{2}\right)^{\log n} = \frac{3^{\log n}}{2^{\log n}}$$