3. **Divide & Conquer Multiplication (w/o Gauss’s Trick)**

\[ a = 4 \quad (4 \text{ recursive calls}) \]
\[ b = 2 \quad (\text{new problems } \frac{1}{2} \text{ size}) \]
\[ d = 1 \quad (\text{constant number of } n\text{-bit additions}) \]

\[ T(1) = \text{constant} \]
\[ b^d = 2^1 = 2 < 4 = a \quad \Rightarrow \text{Case 3} \]
\[ T(n) = O(n^{\log_b a}) = O(n^{\log_2 4}) = O(n^2) \]

4. **Karatsuba Multiplication**

\[ a = 3 \quad (3 \text{ recursive calls}) \]
\[ b = 2 \quad (\text{new problems } \frac{1}{2} \text{ size}) \]
\[ d = 1 \quad (\text{constant # of } n\text{-bit additions}) \]

\[ T(1) = \text{constant} \]
\[ b^d = 2^1 = 2 < 3 = a \quad \Rightarrow \text{Case 3} \]
\[ T(n) = O(n^{\log_b a}) = O(n^{\log_2 3}) = O(n^{1.59}) \]
5. Closest Pair

\[ a = 2 \quad (2 \text{ recursive calls}) \]
\[ b = 2 \quad (n \text{ new problems } \frac{1}{2} \text{ size}) \]
\[ d = 1 \quad (\text{creating } n \text{ new matrices/going through } \frac{1}{2}s) \]

\[ T(3), T(4) \leq \text{constant} \]

\[ b^d = 2^2 = 4 = a \Rightarrow \text{case 1} \]

\[ T(n) = O(n^d \log n) = O(n \log n) \]

\[ \text{If } a \text{ is the number of recursive calls at that level. Don't count recursive calls that happen in other recursive calls.} \]
Proof of Master Method

Q. What is $F$ (in terms of $a$, $b$, $d$)?

A) $\Theta(\log_b n)$  B) $\Theta(\log d n)$  C) $\Theta(n^{\log_b d})$  D) $\Theta(n^{\log_b n})$

Because at each level, problem size is divided by $b$. $\log_b n$ is number of times $n$ can be divided by $b$ before reaching a constant.

$$\text{constant} \cdot b \cdot b \cdots b = n$$

$$\begin{align*}
F \\
\log_b F &= n \\
b^F &= n \\
\frac{n}{c} &= b^F \\
F &= \log_n \left(\frac{n}{c}\right) \\
F &= \log_n n - \log_n c
\end{align*}$$

$$F = \log_b n + \text{constant}$$
Q. What is the total work done at level \( K \) (outside of recursive calls & in terms of \( a, b, d \))?

- \( a^k \) subproblems at level \( K \).
- Level \( K \) subproblem size: \( \left( \frac{n}{b^k} \right)^d \)
- Work outside of recursive call required to solve 1 subproblem

\[ \Rightarrow \text{Total work} \quad a^k \left( \frac{n}{b^k} \right)^d = \left( \frac{a}{b^d} \right)^k n^d \]

Now we add up work done at all levels:

\[
\sum_{k=0}^{\log_b n} \left( \frac{a}{b^d} \right)^k n^d
\]

\[
= n^d \left[ \sum_{k=0}^{\log_b n} \left( \frac{a}{b^d} \right)^k \right]
\]

Multplicative Distributive property

Geometric Series:

\[
\sum_{k=0}^{F} r^k = \begin{cases} 
F+1 & \text{if } r = 1 \\
\frac{1-r^{F+1}}{1-r} & \text{otherwise}
\end{cases}
\]
\[ n^d \left[ \sum_{k=0}^{\log_b n} \left( \frac{a}{b^d} \right)^k \right] = \begin{cases} 
 a = b^d & \rightarrow n^d \left( \log_b n + 1 \right) \rightarrow O(n^d \log_b n) \\
 a > b^d & \rightarrow n^d \left( \frac{1 - \left( \frac{a}{b^d} \right)^n}{1 - \left( \frac{a}{b^d} \right)} \right) \rightarrow O(n^d) \\
 a < b^d & \rightarrow n^d \left( 1 - \left( \frac{a}{b^d} \right)^n \right) \rightarrow O(n^d) \\
 \text{constant} \end{cases} \]

If \( a < b^d \), then \( \frac{a}{b^d} < 1 \), so

\[ 1 - \left( \frac{a}{b^d} \right)^{\log_b n + 1} \leq \left( \frac{a}{b^d} \right)^{\log_b n + 1} < 1 \]

So

\[ \frac{1 - \left( \frac{a}{b^d} \right)^{\log_b n + 1}}{1 - \frac{a}{b^d}} \leq \text{constant} \]

If \( a > b^d \) then \( \frac{a}{b^d} > 1 \), so

\[ 1 - \left( \frac{a}{b^d} \right)^{\log_b n + 1} \leq \frac{\left( \frac{a}{b^d} \right)^{\log_b n + 1} - 1}{\frac{a}{b^d} - 1} \leq \frac{\left( \frac{a}{b^d} \right)^{\log_b n + 1}}{\frac{a}{b^d} - 1} \]

\[ \frac{\left( \frac{a}{b^d} \right)^{\log_b n + 1}}{\frac{a}{b^d} - 1} \leq \text{constant} \]

\[ \left( \frac{a}{b^d} \right)^{\log_b n + 1} = \frac{a^{\log_b n + 1}}{b^{(\log_b n + 1) \log_b d}} = \frac{a^{\log_b n + 1}}{(b^{\log_b d})^{\log_b n}} = \frac{a^{\log_b n}}{(b^{\log_b d})^{\log_b n}} = n^{\log_b a} = n \log_b a \]