CSCI 5421 - Assn. 4
Kelsey Neis, neis@umn.edu

4.2 exercises

4.2-2 Pseudocode for Strassen's

① Divide A, B and C into 4 $n/2 \times n/2$ matrices each $\Theta(1)$

A_{11} = A[1 to n/2][1 to n/2]

A_{12} = A[1 to n/2][n/2 to n]

A_{21} = A[n/2 to n][1 to n/2]

A_{22} = A[n/2 to n][n/2 to n]

⋮

② Create 10 $n/2 \times n/2$ matrices $S_1, S_2, \ldots, S_{10}$, which are the sum or difference of 2 matrices created in step ① $\Theta(n^2)$

$$
\begin{aligned}
S_1 &= B_{12} - B_{22}, \\
S_2 &= A_{11} + A_{12}, \\
S_3 &= A_{21} + A_{22},
\end{aligned}
$$

coded as original

$$
\begin{aligned}
S_2 &= A_{11} + A_{12}, \\
S_3 &= A_{21} + A_{22}, \\
S_4 &= B_{21} - B_{11}, \\
S_5 &= A_{11} + A_{22}, \\
S_6 &= B_{11} + B_{22}, \\
S_7 &= A_{12} - A_{22}, \\
S_8 &= B_{21} + B_{22}, \\
S_9 &= A_{11} - A_{21}, \\
S_{10} &= B_{11} + B_{12}.
\end{aligned}
$$

original matrices, indexed using ranges shown above

③ Recursively compute 7 matrix products, $P_1, P_2, \ldots, P_7$ from $S_1, \ldots, S_{10}$ and matrices of step ① P_i is $n/2 \times n/2$

$$
\begin{aligned}
P_1 &= A_{11} \cdot S_1 &&= A_{11} \cdot B_{12} - A_{11} \cdot B_{22}, \\
P_2 &= S_2 \cdot B_{22} &&= A_{11} \cdot B_{22} + A_{12} \cdot B_{22}, \\
P_3 &= S_3 \cdot B_{11} &&= A_{21} \cdot B_{11} + A_{22} \cdot B_{11}, \\
P_4 &= A_{22} \cdot S_4 &&= A_{22} \cdot B_{21} - A_{22} \cdot B_{11}, \\
P_5 &= S_5 \cdot S_6 &&= A_{11} \cdot B_{11} + A_{11} \cdot B_{22} + A_{22} \cdot B_{11} + A_{22} \cdot B_{22}, \\
P_6 &= S_7 \cdot S_8 &&= A_{12} \cdot B_{21} + A_{12} \cdot B_{22} - A_{22} \cdot B_{21} - A_{22} \cdot B_{22}, \\
P_7 &= S_9 \cdot S_{10} &&= A_{11} \cdot B_{11} + A_{11} \cdot B_{12} - A_{21} \cdot B_{11} - A_{21} \cdot B_{12}.
\end{aligned}
$$

$P_1 = \text{Strassen}(A_{11}, S_1)$

$P_2 = \text{Strassen}(S_2, B_{22})$

$\vdots$

$P_7 = \text{Strassen}(S_9, S_{10})$

④ Compute $C_{11}, C_{12}, C_{21}, C_{22}$ by adding and substracting adding and subtracting various comb'ns of P_i matrices $\Theta(n^2)$

$$C_{11} = P_5 + P_4 - P_2 + P_6$$

$$C_{12} = P_1 + P_2$$

$$C_{21} = P_3 + P_4$$

$$C_{22} = P_5 + P_1 - P_3 - P_7$$

Strassen(A, B)

$n := A.\text{rows}$
let C be $n \times n$ matrix
if $n == 1$, then $C_{11} = A_{11} \cdot B_{11}$
else:
$S_1 = B_{12} - B_{22}$ // where B_{12} is shorthand for
$S_2 = A_{11} + A_{12}$ // the ranged indexes of B,
$S_3 = A_{21} + A_{22}$ // outlined above in (1)
$S_4 = B_{21} - B_{11}$
$S_5 = A_{11} + A_{22}$
$S_6 = B_{11} + B_{22}$
$S_7 = A_{12} - A_{22}$
$S_8 = B_{21} + B_{22}$
$S_9 = A_{11} - A_{21}$
$S_{10} = B_{11} +$

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}$$

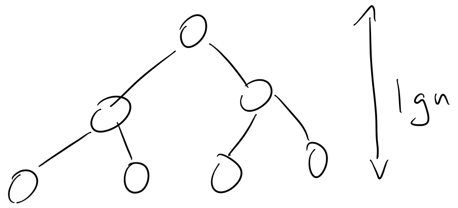
```
        S10 = B11 + B12
        P1 = Strassen(A11, S1)
        P2 = Strassen(S2, B22)
        P3 = Strassen(S3, B11)
        P4 = Strassen(A22, S4)
        P5 = Strassen(S5, S6)
        P6 = Strassen(S7, S8)
        P7 = Strassen(S9, S10)

        C11 = P5 + P4 - P2 + P6
        C12 = P1 + P2
        C21 = P3 + P4
        C22 = P5 + P1 - P3 - P7

    return C
```

4.2-3 modify for n not exact power of 2

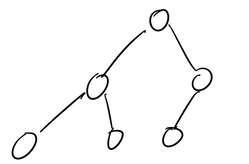
The strassen's algorithm which assumes $n = 2^k$



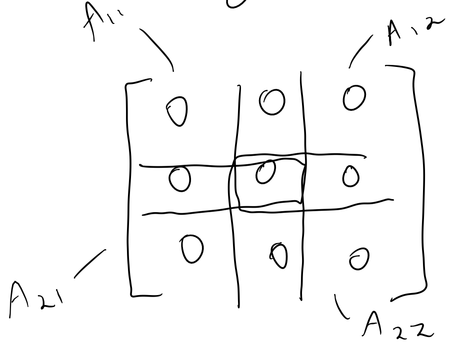
4 x 4 matrix

forming a perfect binary recursion tree, has recursion depth $\lg_2 n$.

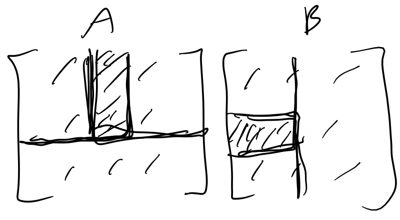
If $n \neq 2^k$, assuming we still divide into 2 subproblems, we get an incomplete tree, for example:



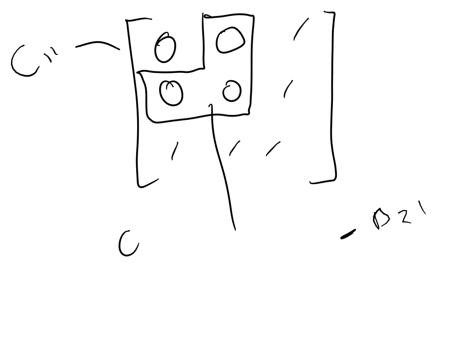
3 X 3 matrix



Since we are assuming that the recurrence will be the same, we still need 7 P matrices, but in this case, the calculation of C needs to take into account the overlap from splitting the matrices into 4 $n/2 \times n/2$ matrices.



As shown in this example of computing C_{11}, The shaded areas represent the overlap of additions that needs to be subtracted.



After making the recursive calls to Strassen, subtract

$C \quad -A_{11}$ calls to Strassen, subtract the opposing quadrant from each C_{11}, C_{12}, C_{21}, and C_{22} to remove the duplication.

Runtime:

Since we still have 7 recursive calls, and splitting into 2 subproblems, we will have the same recursion depth.
We should only be adding 4 arithmetic operations, and the runtime should be the same.

4.2-4 most K multiplications for 3x3 matrices
$\rightarrow O(n^{\lg 7})$ for $n \times n$

Using recurrence for Strassen's:

$$T(n) = 7T(n/2) + \Theta(n^2)$$

$$\downarrow$$

$$K\,T(n/3) + \Theta(n^2)$$

$a = K$, $b = 3$, $n = 3$

Find K where case 1 of the master theorem results in $\leq \Theta(n^{\log 7})$:

$$n^{\log_3 k} \le n^{\log_2 7}$$

$$3^{\log_3 k} \le 3^{\log_2 7}$$

$$k \le 21.849 \rightarrow 21$$

$$\boxed{\begin{aligned} k &= 21 \\ T(n) &= \Theta(n^{\log_3 21}) \end{aligned}}$$

4.2-5

① $132,464 \cdot T(n/68) + n^2$

$a = 132,464 \quad b = 68$

Case 1:

$$n^2 = O(n^{\log_2 132,464 - \epsilon})$$

$$T(n) = \Theta(n^{\log_{68} 132,464}) = n^{2.795128}$$

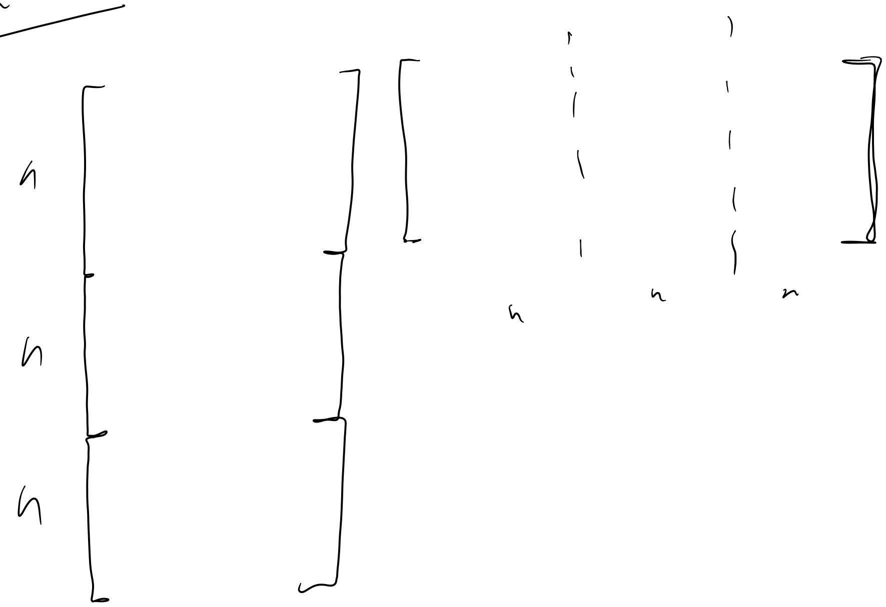
② $143,640 \cdot T(n/70) + n^2$

$$T(n) = \Theta(n^{\log_{72} 143,640}) = n^{2.795122}$$

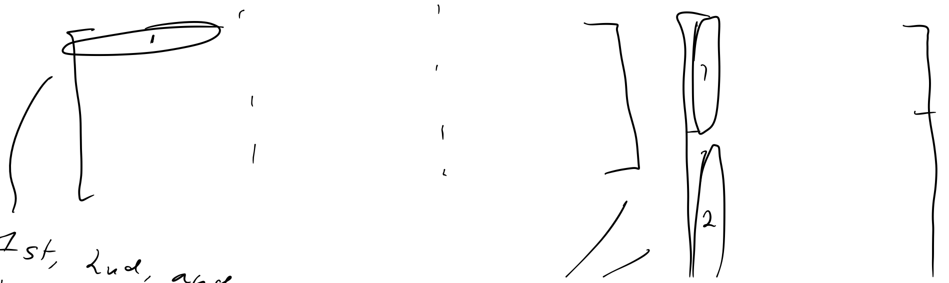
③ $155,424\, T(n/72) + n^2 = n^{2.79514}$

V. Pan's 70 X 70 matrix multiplication algorsthm has the fustest runtime (which is $\sim n^{2.8}$)

4.2-6

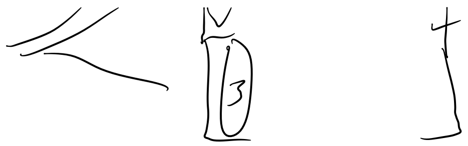


Strassen's can be run on each kth section of the matrices, since the dimensions allow for that here. Runtime will be $\Theta(k \log 7)$.



1st, 2nd, and Strassen

Strassen's, and 3rd call to



To do the reverse, each kth of the matrices on the left would be multiplied by each kth matrix on the right, using the Strassen algorithm. So, this would take $\Theta(k^2 n^{\lg 7})$ time.

4.2-7

$$(a + bi)(c + di)$$

$$ac + bic + adi - bd$$

$$ac - bd \quad + \quad i(bc + ad)$$

$S_1 = (a+b)(c+d) \quad S_2 = ac \quad S_3 = bd$

$$\boxed{\begin{aligned} C_1 &= S_2 - S_3 \\ C_2 &= S_1 - S_2 - S_3 \end{aligned}}$$

4.5 - 3 show binary search recurrence

$T(n) = T(n/2) + \Theta(1)$ is $T(n) = \Theta(\lg n)$

$a = 1 \quad b = 2 \quad f(n) = 1$

Case 1: $1 = O(n^{\log_2 1 - \epsilon})$? NO

Case 2: $1 = \Theta(n^{\log_2 1})$?

$n^{\log_2 1} = 1$ Yes

Case 2 is met, so

$T(n) = \Theta(n^{\log_2 1} \lg n) = \Theta(\lg n)$

4.5-4 Since $f(n)$ is so large, I'm going to start by checking the conditions of case 3:

$a = 4, \quad b = 2, \quad f(n) = n^2 \lg n$

$n^2 \lg n = \Omega(n^{\log_2 1 + \epsilon})$ ✓

$4 f(n/2) \le c f(n), \quad c < 1$

$4 (\frac{n}{2})^2 \lg \frac{n}{2} \le c \cdot n^2 \lg n$

$4\left(\frac{n}{2}\right)^2 \lg \frac{n}{2} \le c \cdot n^2 \lg n$

$n^2[\lg n - 1] \le c \cdot n^2 \lg n$ ✓

Is $f(n)$ polynomially larger than $n^{\log_2 4}$, by a factor of n^{ϵ}?

Yes, $n^2 \lg n / 1 \ge n^2$, and is therefore asymptotically larger, and the Master Theorem can be used.

The upper bound $T(n) = \Theta(n^2 \lg n)$