Homework #1

(Due: March 21)

Task 1. [100 Points] Generalized $\binom{n}{r}$[1]

We are familiar with the following recurrence that computes $\binom{n}{r}$ representing the number of ways one can choose r distinct objects from n distinct objects, where $n \geq 0$ and r are integers.

$$\binom{n}{r} = \begin{cases} 0 & \text{if } r < 0 \text{ or } r > n, \\ 1 & \text{if } n = r = 0, \\ \binom{n-1}{r-1} + \binom{n-1}{r} & \text{otherwise.} \end{cases}$$

The following recurrence defines a more general function $S(n, r)$ for integers $n \geq 0$ and r, and real-valued constants a and b.

$$S(n,r) = \begin{cases} 0 & \text{if } r < 0 \text{ or } r > n, \\ 1 & \text{if } n = r = 0, \\ a \cdot S(n-1, r-1) + b \cdot S(n-1, r) & \text{otherwise.} \end{cases}$$

Now design an algorithm that given the values of n, a and b, computes $S(n, r)$ for all $r \in [0, n]$ in $\mathcal{O}(n \log n)$ time without using any closed-form solution for $S(n, r)$.

Task 2. [100 Points] Special Case to General Case

The following recurrence defines a function $H(n, r)$ for integers $N > 0$, $n \geq 0$, $r \in [0, N + 1]$, and real-valued constants a, b, c, α, β, $\gamma_1, \gamma_2, \ldots, \gamma_N$.

$$H(n,r) = \begin{cases} \alpha & \text{if } r = 0, \\ \beta & \text{if } r = N+1, \\ \gamma_r & \text{if } n = 0 \text{ and } 1 \leq r \leq N, \\ a \cdot H(n-1, r-1) + b \cdot H(n-1, r) + c \cdot H(n-1, r+1) & \text{otherwise.} \end{cases}$$

There exists an algorithm, let's call it Comp-Boundary-Zero, that for any given $N \in \mathbb{N}$ and $N' \in [1, N]$ can compute $H(N', r)$ for all $r \in [1, N]$ in $\mathcal{O}(N \log N)$ time provided $\alpha = \beta = 0$.

Design a recursive divide-and-conquer algorithm that uses Comp-Boundary-Zero as a subroutine to compute $H(N, r)$ for all $r \in [1, N]$ in $\mathcal{O}\left(N \log^2 N\right)$ time without imposing any constraints on α and β. Write a recurrence relation describing the running time of your algorithm and solve it.

[Optional: No Point] Design an algorithm (Comp-Boundary-Zero) that for any given $N \in \mathbb{N}$ and $N' \in [1, N]$ can compute $H(N', r)$ for all $r \in [1, N]$ in $\mathcal{O}(N \log N)$ time provided $\alpha = \beta = 0$.

[1] Original idea by Pramod Ganapathi

Input: $n \times n$ matrices X and Y
Output: $Z \leftarrow XY$

1. ***if*** X is a 1×1 matrix ***then*** $Z \leftarrow XY$
2. ***else***

   { Compute input matrices } ....

3. $X_0 \leftarrow X_{12} + X_{21}$, $Y_0 \leftarrow Y_{12} + Y_{21}$,
   $X_1 \leftarrow X_{21} + X_{22}$, $Y_1 \leftarrow Y_{21}$,
   $X_2 \leftarrow X_{12}$, $Y_2 \leftarrow Y_{22} - Y_{12}$,
   $X_3 \leftarrow X_{21}$, $Y_3 \leftarrow Y_{11} - Y_{21}$,
   $X_4 \leftarrow X_{11} + X_{12}$, $Y_4 \leftarrow Y_{12}$,
   $X_5 \leftarrow X_{22} - X_{12}$, $Y_5 \leftarrow Y_{21} + Y_{22}$,
   $X_6 \leftarrow X_{11} - X_{21}$, $Y_6 \leftarrow Y_{11} + Y_{12}$

   { Recursion } ....

4. $P_0 \leftarrow X_0 Y_0$, $P_1 \leftarrow X_1 Y_1$, $P_2 \leftarrow X_2 Y_2$,
   $P_3 \leftarrow X_3 Y_3$, $P_4 \leftarrow X_4 Y_4$, $P_5 \leftarrow X_5 Y_5$,
   $P_6 \leftarrow X_6 Y_6$

   { Compute output matrix } ....

5. $Z_{11} \leftarrow P_0 + P_3 - P_4 + P_6$, $Z_{12} \leftarrow P_2 + P_4$,
   $Z_{21} \leftarrow P_1 + P_3$, $Z_{22} \leftarrow P_0 - P_1 + P_2 + P_5$

Figure 1: [Task 3] A simple variant of Strassen's algorithm.

Task 3. [100 Points] Deriving Strassen's Algorithm

We studied Strassen's matrix multiplication algorithm in Lecture 3. However, the algorithm we derived in slides 17–18 is not exactly the sams as the algorithm we learnt in slides 5–15!

(*a*) [**20 Points**] Write down the algorithm we derived in in slides 17–18. How is it different from Strassen's algorithm we saw in slides 5–15?

(*b*) [**20 Points**] Figure 1 shows a simple variant of Strassen's algorithm. Prove that the variant correctly multiplies two $n \times n$ matrices.

(*c*) [**60 Points**] Derive the algorithm shown in Figure 1. You must use the approach shown in slides 17–18 of Lecture 3.