RECURRENCES

335 Fall 2022 – Recitation 3

Substitution Method

1. $T(n) = T(n-1) + n$ is $O(n^2)$
2. $T(n) = T(n/2) + 1$ is $O(\log(n))$
3. $T(n) = 2T(n/2) + n$ is $O(n\log(n))$

Recursion Tree

1. $T(n) = T(n/2) + n^2$
2. $T(n) = 2T(n-1) + 1$
3. $T(n) = 3T(n/2) + n$
4. $T(n) = T(n/2) + T(n/4) + T(n/8) + n$

Use substitution method to verify your answers

Master Theorem

1. $T(n) = 2T(n/4) + 1$
2. $T(n) = 2T(n/4) + \sqrt{n}$
3. $T(n) = 2T(n/4) + n^2$
4. $T(n) = T(7n/10) + n$
5. $T(n) = 7T(n/3) + n^2$
6. $T(n) = 7T(n/2) + n^2$
7. $T(n) = 4T(\sqrt{n}) + \log(n)$

- Professor Caesar wishes to develop a matrix-multiplication algorithm that is asymptotically faster than Strassen's algorithm. His algorithm will use the divide-and-conquer method, dividing each matrix into pieces of size n/4 x n/4, and the divide and combine steps together will take $\Theta(n^2)$ time. He needs to determine how many subproblems his algorithm has to create in order to beat Strassen's algorithm. If his algorithm creates a*a* subproblems, then the recurrence for the running time $T(n)$ becomes $T(n) = aT(n/4) + \Theta(n^2)$. What is the largest integer value of a*a* for which Professor Caesar's algorithm would be asymptotically faster than Strassen's algorithm?