

CS375 Small Assignment 5

Due Oct 9th 10PM ET

Use the recursion tree method to determine the runtime of the following functions. You can use the Master Theorem to check your answer for correctness:

Master Theorem

Suppose $T(n) = aT(\frac{n}{b}) + \Theta(f(n))$:

- a) if there exists an $\epsilon > 0$ such that $f(n) = O(n^{\log_b(a)-\epsilon})$, then $T(n) = \Theta(n^{\log_b(a)})$
- b) if there exists a $k \geq 0$ such that $f(n) = \Theta(n^{\log_b(a)} \log^k n)$, then $T(n) = \Theta(n^{\log_b(a)} \log^{k+1} n)$
- c) if there exists an $\epsilon > 0$ such that $f(n) = \Omega(n^{\log_b(a)+\epsilon})$ and there exists a $c < 1$ such that $af(\frac{n}{b}) < cf(n)$, then $T(n) = \Theta(f(n))$.

1. $T(n) = 2T(\frac{n}{2}) + \Theta(n^3)$
2. $T(n) = T(9n/10) + \Theta(n)$
3. $T(n) = 16T(\frac{n}{4}) + \Theta(n^2)$