# Announcements

- TA's Office Hours now
  - Monday, 2:00pm – 2:30pm
  - Wednesday, 2:00pm – 3:30pm
- Reading Assignments:
  Chapter 4 & 5 (Textbook: CLRS)
- Reminder: Homework #1 is due today
- Homework #2 is due on Sept 20, 2005

# Iteration Method

- **Expand (iterate) the recurrence and express it as a summation of terms dependent only on $n$ and the initial conditions**
- **The key is to focus on 2 parameters**
  - **the number of times the recurrence needs to be iterated to reach the boundary condition**
  - **the sum of terms arising from each level of the iteration process**
- **Techniques for evaluating summations can then be used to provide bounds on solution.**

# An Example

- **Solve:** $T(n) = 3T(n/4) + n$

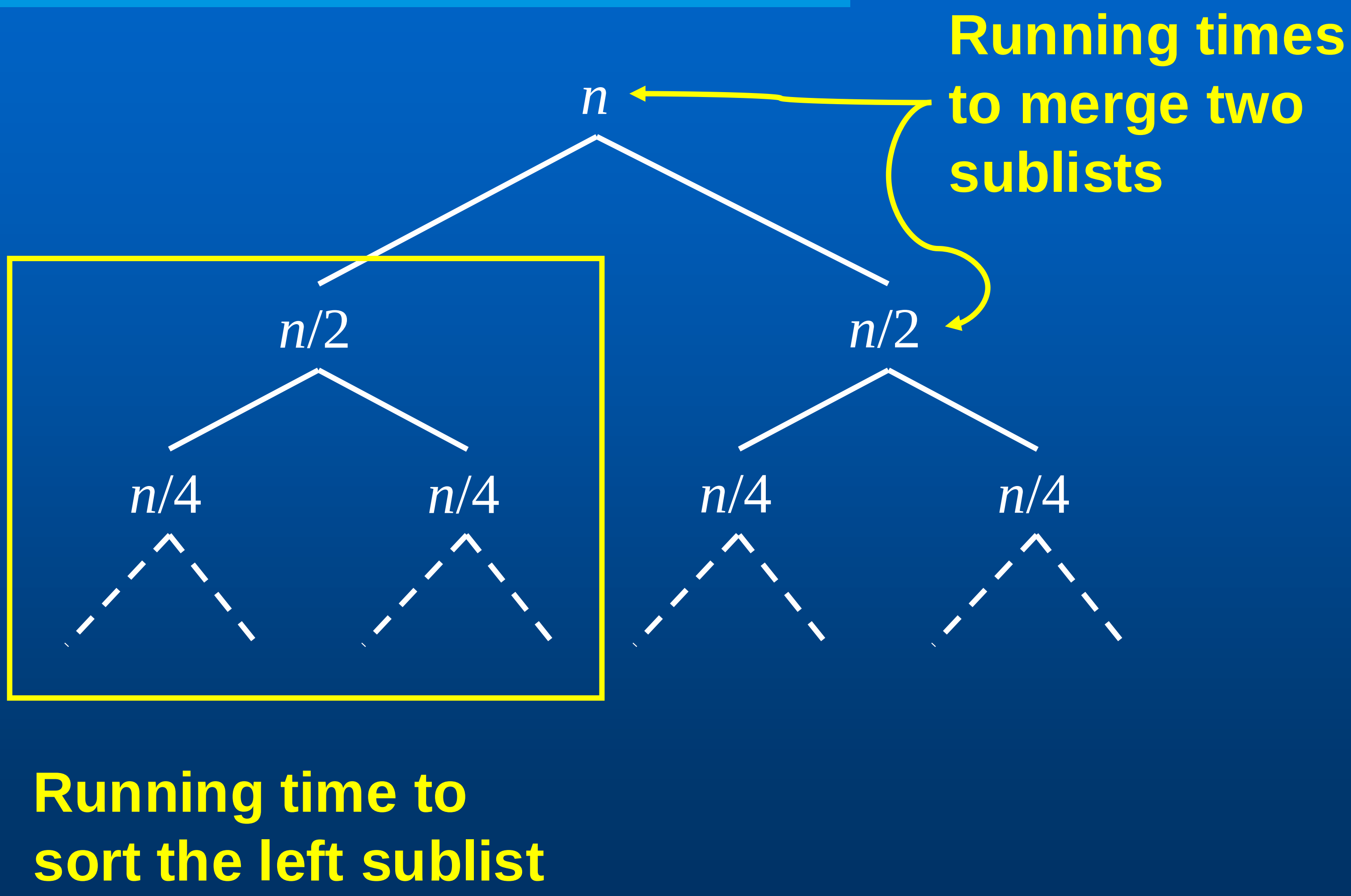
$$T(n) = n + 3T(n/4)$$
$$= n + 3[\, n/4 + 3T(n/16)\, ]$$
$$= n + 3[n/4] + 9T(n/16)$$
$$= n + 3[n/4] + 9\,[n/16] + 27T(n/64)$$

$$T(n) \leq n + 3n/4 + 9n/16 + 27n/64 + \ldots + 3^{\log_4 n}\Theta(1)$$
$$\leq n \sum (3/4)^i + \Theta(n^{\log_4 3})$$
$$= 4n + o(n)$$
$$= O(n)$$

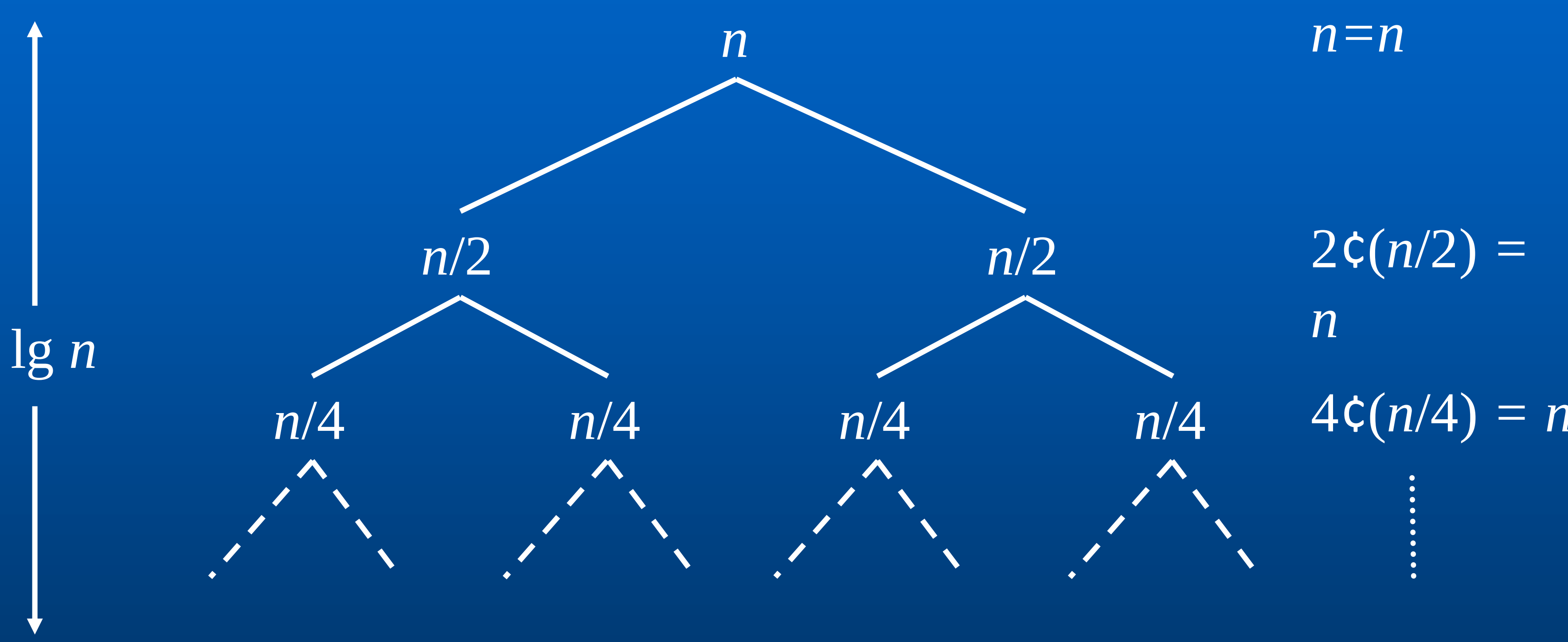
# Recursion Trees

- Keep track of the time spent on the subproblems of a divide and conquer algorithm
- A convenient way to visualize what happens when a recursion is iterated
- Help organize the algebraic bookkeeping necessary to solve the recurrence

# Merge Sort


Running times to merge two sublists
n
n/2
n/2
n/4
n/4
n/4
n/4
Running time to sort the left sublist

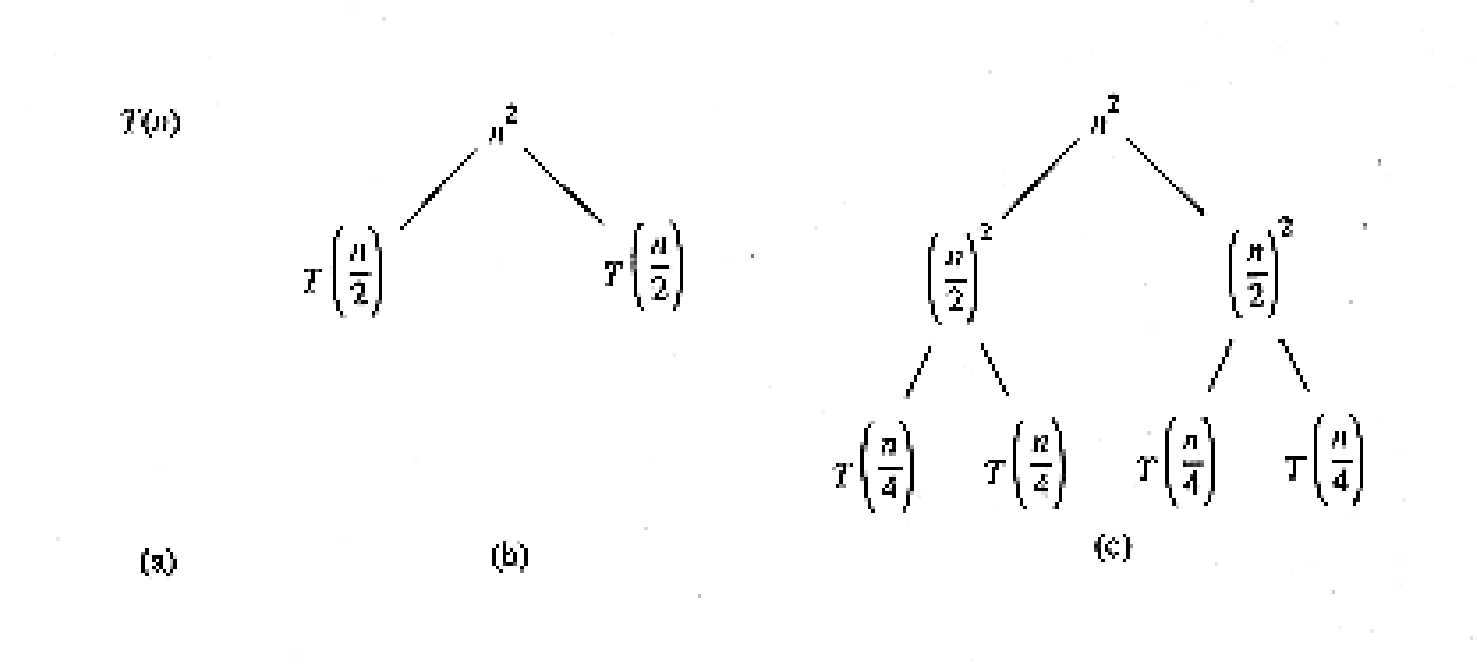
# Running Time



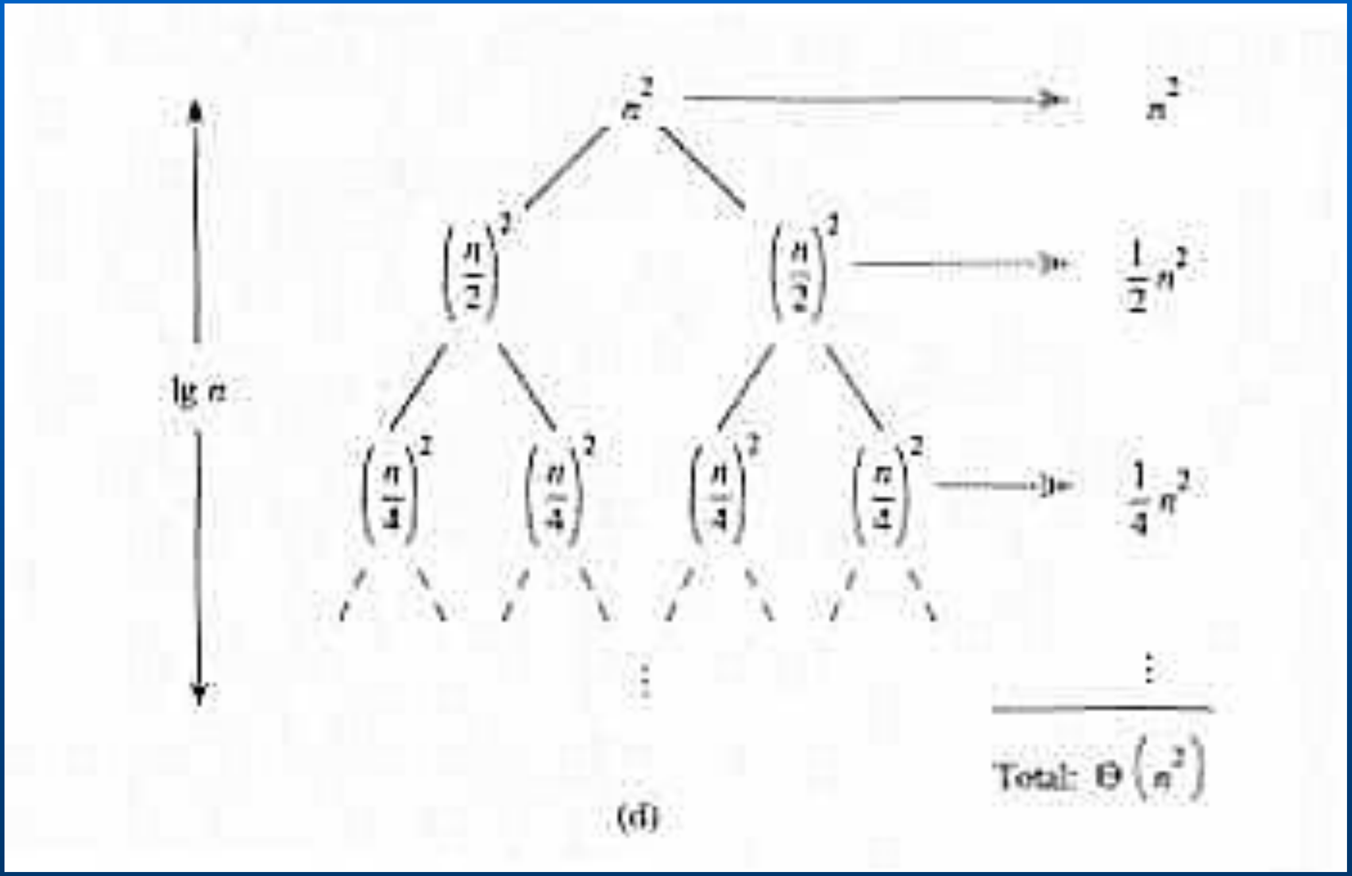

**Total:** $n \lg n$

# Recursion Trees and Recurrences

- Useful even when a specific algorithm is not specified
  - For $T(n) = 2T(n/2) + n^2$, we have



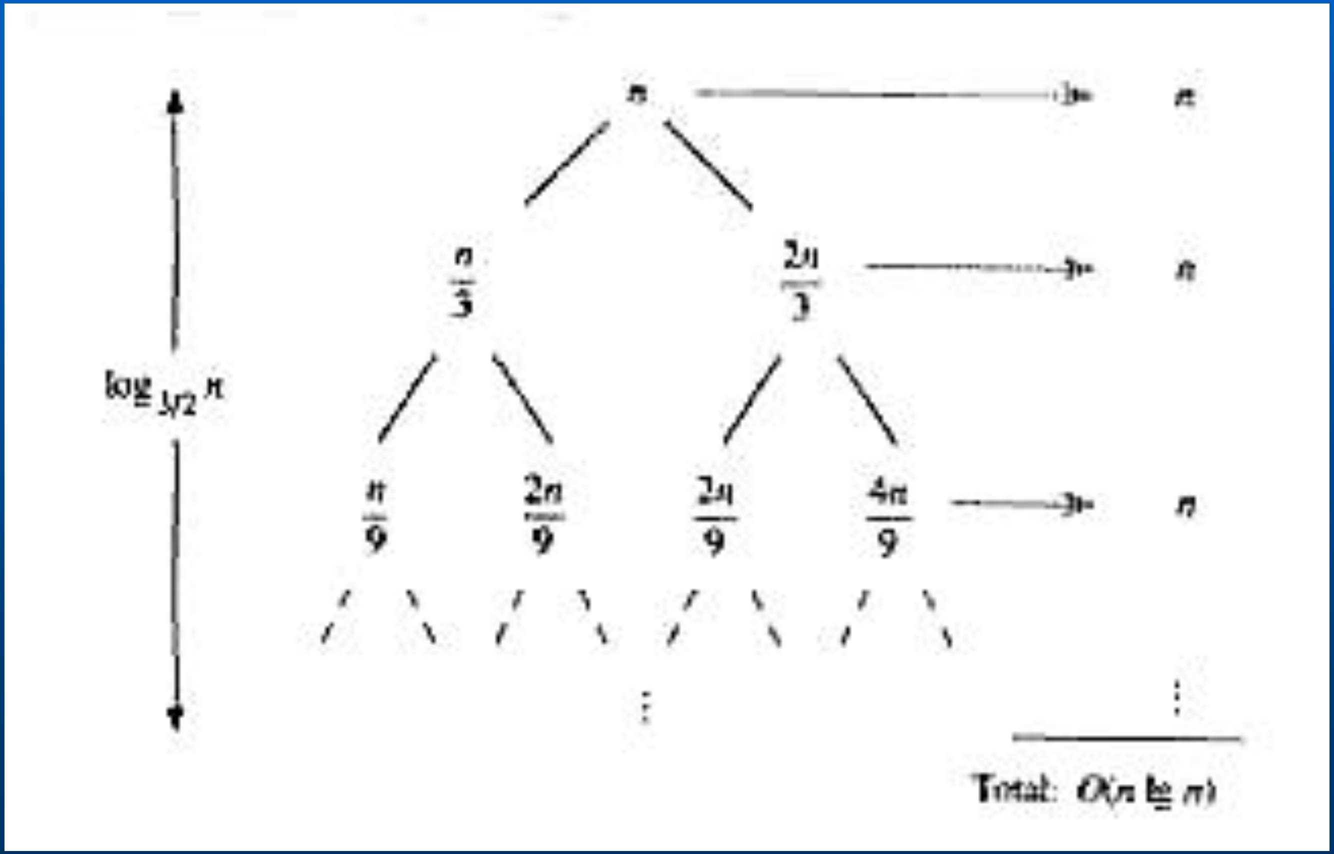
# Recursion Trees




$$T(n) = \Theta(n^2)$$

# Recursion Trees

- For $T(n) = T(n/3) + T(2n/3) + n$




$$T(n) = O(n \lg n)$$

# Master Method

- Provides a "cookbook" method for solving recurrences of the form

$$T(n) = a\,T(n/b) + f(n)$$

- Assumptions:
  - $a \geq 1$ and $b \geq 1$ are constants
  - $f(n)$ is an asymptotically positive function
  - $T(n)$ is defined for nonnegative integers
  - We interpret $n/b$ to mean either $\lfloor n/b \rfloor$ or $\lceil n/b \rceil$

# The Master Theorem

With the recurrence $T(n) = a\,T(n/b) + f(n)$ as in the previous slide, $T(n)$ can be bounded asymptotically as follows:

1. If $f(n)=O(n^{\log_b a-\varepsilon})$ for some constant $\varepsilon > 0$, then $T(n)= \Theta(n^{\log_b a})$.
2. If $f(n) = \Theta(n^{\log_b a})$, then $T(n) = \Theta(n^{\log_b a} \lg n)$.
3. If $f(n) = \Omega( n^{\log_b a+\varepsilon})$ for some constant $\varepsilon > 0$, and if $a\,f(n/b) \le c\,f(n)$ for some constant $c < 1$ and all sufficiently large $n$, then $T(n)= \Theta(f(n))$.

# Simplified Master Theorem

Let $a \geq 1$ and $b > 1$ be constants and let $T(n)$ be the recurrence

$$T(n) = a\,T(n/b) + c\,n^k$$

defined for $n \geq 0$.

1. If $a > b^k$, then $T(n) = \Theta(\, n^{\log_b a}\,)$.
2. If $a = b^k$, then $T(n) = \Theta(\, n^k \lg n\,)$.
3. If $a < b^k$, then $T(n) = \Theta(\, n^k\,)$.

# Examples

- $T(n) = 16T(n/4) + n$
  - $a = 16$, $b = 4$, thus $n^{\log_b a} = n^{\log_4 16} = \Theta(n^2)$
  - $f(n) = n = O(n^{\log_4 16 - \varepsilon})$ where $\varepsilon = 1 \Rightarrow$ case 1.
  - Therefore, $T(n) = \Theta(n^{\log_b a}) = \Theta(n^2)$

- $T(n) = T(3n/7) + 1$
  - $a = 1$, $b=7/3$, and $n^{\log_b a} = n^{\log_{7/3} 1} = n^0 = 1$
  - $f(n) = 1 = \Theta(n^{\log_b a}) \Rightarrow$ case 2.
  - Therefore, $T(n) = \Theta(n^{\log_b a} \lg n) = \Theta(\lg n)$

# Examples (Cont.)

- $T(n) = 3T(n/4) + n \lg n$
  - $a = 3$, $b=4$, thus $n^{\log_b a} = n^{\log_4 3} = O(n^{0.793})$
  - $f(n) = n \lg n = \Omega(n^{\log_4 3 + \varepsilon})$ where $\varepsilon \approx 0.2 \Rightarrow$ case 3.
  - Therefore, $T(n) = \Theta(f(n)) = \Theta(n \lg n)$

- $T(n) = 2T(n/2) + n \lg n$
  - $a = 2$, $b=2$, $f(n) = n \lg n$, and $n^{\log_b a} = n^{\log_2 2} = n$
  - $f(n)$ is asymptotically larger than $n^{\log_b a}$, but not polynomially larger. The ratio $\lg n$ is asymptotically less than $n^{\varepsilon}$ for any positive $\varepsilon$. Thus, the Master Theorem *doesn't* apply here.

# Exercises

- Use the Master Method to solve the following:
  1. $T(n) = 4T(n/2) + n$
  2. $T(n) = 4T(n/2) + n^2$
  3. $T(n) = 4T(n/2) + n^3$