



CS201 DISCRETE MATHEMATICS FOR COMPUTER SCIENCE

Dr. QI WANG

Department of Computer Science and Engineering

Office: Room903, Nanshan iPark A7 Building

Email: wangqi@sustech.edu.cn

Recurrences

- A *recurrence equation* or *recurrence* for a function defined on the set of integers $\geq b$ is one that tells us how to compute the n th value from some or all the first $n - 1$ values.



Recurrences

- A *recurrence equation* or *recurrence* for a function defined on the set of integers $\geq b$ is one that tells us how to compute the n th value from some or all the first $n - 1$ values.

To completely specify a function on the basis of a recurrence, we have to give the *initial condition(s)* (a.k.a. the *base case(s)*) for the recurrence.



Recurrences

- A *recurrence equation* or *recurrence* for a function defined on the set of integers $\geq b$ is one that tells us how to compute the n th value from some or all the first $n - 1$ values.

To completely specify a function on the basis of a recurrence, we have to give the *initial condition(s)* (a.k.a. the *base case(s)*) for the recurrence.

$$M(n) = \begin{cases} 1 & \text{if } n = 1 \\ 2M(n - 1) + 1 & \text{otherwise} \end{cases} \quad \text{Towers of Hanoi}$$

$$F(n) = \begin{cases} 1 & \text{if } n = 0, 1 \\ F(n - 1) + F(n - 2) & \text{otherwise} \end{cases} \quad \text{Fibonacci Sequence}$$



Recurrences

- **Example 2:** Let $S(n)$ be the number of subsets of a set of size n . What is the formula for $S(n)$?

The empty set, of size $n = 0$ has only one subset (itself), so $S(0) = 1$.

It is not difficult to see that

$$S(1) = 2, S(2) = 4, S(3) = 8$$



Recurrences

- **Example 2:** Let $S(n)$ be the number of subsets of a set of size n . What is the formula for $S(n)$?

The empty set, of size $n = 0$ has only one subset (itself), so $S(0) = 1$.

It is not difficult to see that

$$S(1) = 2, S(2) = 4, S(3) = 8$$

We “guess” that $S(n) = 2^n$. But, in order to prove formula, we’ll need to think recursively.



Recurrences

- Consider the eight subsets of $\{1, 2, 3\}$:
 $\emptyset, \{1\}, \{2\}, \{1, 2\}, \{3\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}$



Recurrences

- Consider the eight subsets of $\{1, 2, 3\}$:

$\emptyset, \{1\}, \{2\}, \{1, 2\}, \{3\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}$

\emptyset	$\{1\}$	$\{2\}$	$\{1, 2\}$
$\{3\}$	$\{1, 3\}$	$\{2, 3\}$	$\{1, 2, 3\}$



Recurrences

- Consider the eight subsets of $\{1, 2, 3\}$:

$\emptyset, \{1\}, \{2\}, \{1, 2\}, \{3\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}$

\emptyset	$\{1\}$	$\{2\}$	$\{1, 2\}$
$\{3\}$	$\{1, 3\}$	$\{2, 3\}$	$\{1, 2, 3\}$

First four subsets are exactly the subsets of $\{1, 2\}$, while second four are the subsets of $\{1, 2\}$ with 3 added into each.



Recurrences

- Consider the eight subsets of $\{1, 2, 3\}$:

$\emptyset, \{1\}, \{2\}, \{1, 2\}, \{3\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}$

\emptyset	$\{1\}$	$\{2\}$	$\{1, 2\}$
$\{3\}$	$\{1, 3\}$	$\{2, 3\}$	$\{1, 2, 3\}$

First four subsets are exactly the subsets of $\{1, 2\}$, while second four are the subsets of $\{1, 2\}$ with 3 added into each.

So, we get a subset of $\{1, 2, 3\}$ either by taking a subset of $\{1, 2\}$ or by adjoining 3 to a subset of $\{1, 2\}$.



Recurrences

- Consider the eight subsets of $\{1, 2, 3\}$:

$\emptyset, \{1\}, \{2\}, \{1, 2\}, \{3\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}$

\emptyset	$\{1\}$	$\{2\}$	$\{1, 2\}$
$\{3\}$	$\{1, 3\}$	$\{2, 3\}$	$\{1, 2, 3\}$

First four subsets are exactly the subsets of $\{1, 2\}$, while second four are the subsets of $\{1, 2\}$ with 3 added into each.

So, we get a subset of $\{1, 2, 3\}$ either by taking a subset of $\{1, 2\}$ or by adjoining 3 to a subset of $\{1, 2\}$.

This suggests that the recurrence for the number of subsets of an n -element set $\{1, 2, \dots, n\}$ is

$$S(n) = \begin{cases} 1 & \text{if } n = 0 \\ 2S(n-1) & \text{if } n \geq 1 \end{cases}$$



Recurrences

- **Proof.** of correctness of this recurrence



Recurrences

- **Proof.** of correctness of this recurrence

The subsets of $\{1, 2, \dots, n\}$ can be partitioned according to whether or not they contain the element n .



Recurrences

■ **Proof.** of correctness of this recurrence

The subsets of $\{1, 2, \dots, n\}$ can be partitioned according to whether or not they contain the element n .

Each subset S containing n can be constructed in a unique fashion by adding n to the subset $S - \{n\}$ not containing n .

Each subset S not containing n can be constructed by removing n from the unique set $S \cup \{n\}$ containing n .



Recurrences

■ **Proof.** of correctness of this recurrence

The subsets of $\{1, 2, \dots, n\}$ can be partitioned according to whether or not they contain the element n .

Each subset S containing n can be constructed in a unique fashion by adding n to the subset $S - \{n\}$ not containing n .

Each subset S not containing n can be constructed by removing n from the unique set $S \cup \{n\}$ containing n .

So, the number of subsets containing n is exactly the same as the number of subsets not containing n .



Recurrences

■ **Proof.** of correctness of this recurrence

The subsets of $\{1, 2, \dots, n\}$ can be partitioned according to whether or not they contain the element n .

Each subset S containing n can be constructed in a unique fashion by adding n to the subset $S - \{n\}$ not containing n .

Each subset S not containing n can be constructed by removing n from the unique set $S \cup \{n\}$ containing n .

So, the number of subsets containing n is exactly the same as the number of subsets not containing n .

Thus, if $n > 1$, then $S(n) = 2S(n - 1)$.



Recurrences

■ **Proof.** of correctness of this recurrence

The subsets of $\{1, 2, \dots, n\}$ can be partitioned according to whether or not they contain the element n .

Each subset S containing n can be constructed in a unique fashion by adding n to the subset $S - \{n\}$ not containing n .

Each subset S not containing n can be constructed by removing n from the unique set $S \cup \{n\}$ containing n .

So, the number of subsets containing n is exactly the same as the number of subsets not containing n .

Thus, if $n > 1$, then $S(n) = 2S(n - 1)$.

Proof by induction is easy.



Iterating a Recurrence

- Let $T(n) = rT(n-1) + a$,
where r and a are constants.



Iterating a Recurrence

- Let $T(n) = rT(n-1) + a$,
where r and a are constants.

Find a **recurrence** that expresses

$T(n)$ in terms of $T(n-2)$

$T(n)$ in terms of $T(n-3)$

$T(n)$ in terms of $T(n-4)$

⋮



Iterating a Recurrence

- Let $T(n) = rT(n-1) + a$,
where r and a are constants.

Find a **recurrence** that expresses

$T(n)$ in terms of $T(n-2)$

$T(n)$ in terms of $T(n-3)$

$T(n)$ in terms of $T(n-4)$

⋮

Can we generalize this to find a **closed-form solution**?



Iterating a Recurrence

- Note that $T(n) = rT(n-1) + a$ implies that
 $\forall i < n, T(n-i) = rT((n-i)-1) + a$



Iterating a Recurrence

- Note that $T(n) = rT(n-1) + a$ implies that

$$\forall i < n, T(n-i) = rT((n-i)-1) + a$$

Then, we have

$$\begin{aligned} T(n) &= rT(n-1) + a \\ &= r(rT(n-2) + a) + a \\ &= r^2 T(n-2) + ra + a \\ &= r^2(rT(n-3) + a) + ra + a \\ &= r^3 T(n-3) + r^2 a + ra + a \\ &= r^3(rT(n-4) + a) + r^2 a + ra + a \\ &= r^4 T(n-4) + r^3 a + r^2 a + ra + a. \end{aligned}$$



Iterating a Recurrence

- Note that $T(n) = rT(n-1) + a$ implies that

$$\forall i < n, T(n-i) = rT((n-i)-1) + a$$

Then, we have

$$\begin{aligned} T(n) &= rT(n-1) + a \\ &= r(rT(n-2) + a) + a \\ &= r^2 T(n-2) + ra + a \\ &= r^2(rT(n-3) + a) + ra + a \\ &= r^3 T(n-3) + r^2 a + ra + a \\ &= r^3(rT(n-4) + a) + r^2 a + ra + a \\ &= r^4 T(n-4) + r^3 a + r^2 a + ra + a. \end{aligned}$$

Guess $T(n) = r^n T(0) + a \sum_{i=0}^{n-1} r^i$



Iterating a Recurrence

- The method we used to guess the solution is called *iterating the recurrence*, because we repeatedly (iteratively) use the recurrence.



Iterating a Recurrence

- The method we used to guess the solution is called *iterating the recurrence*, because we repeatedly (iteratively) use the recurrence.

Another approach is to iterate from the “bottom-up” instead of “top-down”.



Iterating a Recurrence

- The method we used to guess the solution is called *iterating the recurrence*, because we repeatedly (iteratively) use the recurrence.

Another approach is to iterate from the “bottom-up” instead of “top-down”.

$$T(0) = b$$

$$T(1) = rT(0) + a = rb + a$$

$$T(2) = rT(1) + a = r(rb + a) + a = r^2b + ra + a$$

$$T(3) = rT(2) + a = r^3b + r^2a + ra + a$$



Iterating a Recurrence

- The method we used to guess the solution is called *iterating the recurrence*, because we repeatedly (iteratively) use the recurrence.

Another approach is to iterate from the “bottom-up” instead of “top-down”.

$$T(0) = b$$

$$T(1) = rT(0) + a = rb + a$$

$$T(2) = rT(1) + a = r(rb + a) + a = r^2b + ra + a$$

$$T(3) = rT(2) + a = r^3b + r^2a + ra + a$$

This would lead to the same guess

$$T(n) = r^n b + a \sum_{i=0}^{n-1} r^i.$$



Formula of Recurrences

- **Theorem** If $T(n) = rT(n-1) + a$, $T(0) = b$, and $r \neq 1$, then

$$T(n) = r^n b + a \frac{1 - r^n}{1 - r}$$

for all nonnegative integers n .



Formula of Recurrences

- **Theorem** If $T(n) = rT(n-1) + a$, $T(0) = b$, and $r \neq 1$, then

$$T(n) = r^n b + a \frac{1 - r^n}{1 - r}$$

for all nonnegative integers n .

Proof by induction

The base case:

$$T(0) = r^0 b + a \frac{1 - r^0}{1 - r} = b.$$

So the formula is true when $n = 0$.

Now assume that $n > 0$ and

$$T(n-1) = r^{n-1} b + a \frac{1 - r^{n-1}}{1 - r}.$$



Formula of Recurrences

■ Proof by induction

$$\begin{aligned}T(n) &= rT(n-1) + a \\&= r \left(r^{n-1}b + a \frac{1-r^{n-1}}{1-r} \right) + a \\&= r^n b + \frac{ar - ar^n}{1-r} + a \\&= r^n b + \frac{ar - ar^n + a - ar}{1-r} \\&= r^n b + a \frac{1-r^n}{1-r}.\end{aligned}$$



Formula of Recurrences

- **Theorem** If $T(n) = rT(n-1) + a$, $T(0) = b$, and $r \neq 1$, then

$$T(n) = r^n b + a \frac{1 - r^n}{1 - r}$$

for all nonnegative integers n .



Formula of Recurrences

- **Theorem** If $T(n) = rT(n-1) + a$, $T(0) = b$, and $r \neq 1$, then

$$T(n) = r^n b + a \frac{1 - r^n}{1 - r}$$

for all nonnegative integers n .

Example:

$$T(n) = 3T(n-1) + 2 \text{ with } T(0) = 5$$



Formula of Recurrences

- **Theorem** If $T(n) = rT(n-1) + a$, $T(0) = b$, and $r \neq 1$, then

$$T(n) = r^n b + a \frac{1 - r^n}{1 - r}$$

for all nonnegative integers n .

Example:

$$T(n) = 3T(n-1) + 2 \text{ with } T(0) = 5$$

Plugging $r = 3$, $a = 2$, $b = 5$ in the formula, gives

$$T(n) = 3^n \cdot 5 + 2 \frac{1 - 3^n}{1 - 3} = 3^n \cdot 6 - 1$$



First-Order Linear Recurrences

- A recurrence of the form $T(n) = f(n)T(n-1) + g(n)$ is called a *first-order linear recurrence*.

一阶线性递归



First-Order Linear Recurrences

- A recurrence of the form $T(n) = f(n)T(n-1) + g(n)$ is called a *first-order linear recurrence*.
 - ◇ **First Order** because it only depends upon going back one step, i.e., $T(n-1)$



First-Order Linear Recurrences

- A recurrence of the form $T(n) = f(n)T(n-1) + g(n)$ is called a *first-order linear recurrence*.
 - ◇ **First Order** because it only depends upon going back one step, i.e., $T(n-1)$

If it depends upon $T(n-2)$, it would be a **second-order** recurrence, e.g., $T(n) = T(n-1) + 2T(n-2)$.



First-Order Linear Recurrences

- A recurrence of the form $T(n) = f(n)T(n-1) + g(n)$ is called a *first-order linear recurrence*.
 - ◇ **First Order** because it only depends upon going back one step, i.e., $T(n-1)$

If it depends upon $T(n-2)$, it would be a **second-order** recurrence, e.g., $T(n) = T(n-1) + 2T(n-2)$.
 - ◇ **Linear** because $T(n-1)$ only appears to the **first power**.



First-Order Linear Recurrences

- A recurrence of the form $T(n) = f(n)T(n-1) + g(n)$ is called a *first-order linear recurrence*.
 - ◇ **First Order** because it only depends upon going back one step, i.e., $T(n-1)$

If it depends upon $T(n-2)$, it would be a **second-order** recurrence, e.g., $T(n) = T(n-1) + 2T(n-2)$.
 - ◇ **Linear** because $T(n-1)$ only appears to the **first power**.

Something like $T(n) = (T(n-1))^2 + 3$ would be a **non-linear** first-order recurrence relation.



First-Order Linear Recurrences

- $T(n) = f(n)T(n-1) + g(n)$



First-Order Linear Recurrences

- $T(n) = f(n)T(n-1) + g(n)$

When $f(n)$ is a constant, say r , the general solution is almost as easy as we derived before. Iterating the recurrence gives

$$\begin{aligned}T(n) &= rT(n-1) + g(n) \\&= r(rT(n-2) + g(n-1)) + g(n) \\&= r^2T(n-2) + rg(n-1) + g(n) \\&= r^3T(n-3) + r^2g(n-2) + rg(n-1) + g(n) \\&\vdots \\&= r^nT(0) + \sum_{i=0}^{n-1} r^i g(n-i)\end{aligned}$$



First-Order Linear Recurrences

- **Theorem** For any positive constants a and r , and any function g defined on nonnegative integers, the solution to the first-order linear recurrence

$$T(n) = \begin{cases} rT(n-1) + g(n) & \text{if } n > 0 \\ a & \text{if } n = 0 \end{cases}$$

is

$$T(n) = r^n a + \sum_{i=1}^n r^{n-i} g(i).$$



First-Order Linear Recurrences

- **Theorem** For any positive constants a and r , and any function g defined on nonnegative integers, the solution to the first-order linear recurrence

$$T(n) = \begin{cases} rT(n-1) + g(n) & \text{if } n > 0 \\ a & \text{if } n = 0 \end{cases}$$

is

$$T(n) = r^n a + \sum_{i=1}^n r^{n-i} g(i).$$

Proof by induction



Examples

- Solve $T(n) = 4T(n-1) + 2^n$ with $T(0) = 6$



Examples

- Solve $T(n) = 4T(n-1) + 2^n$ with $T(0) = 6$

$$\begin{aligned}T(n) &= 6 \cdot 4^n + \sum_{i=1}^n 4^{n-i} \cdot 2^i \\&= 6 \cdot 4^n + 4^n \sum_{i=1}^n 4^{-i} \cdot 2^i \\&= 6 \cdot 4^n + 4^n \sum_{i=1}^n \left(\frac{1}{2}\right)^i \\&= 6 \cdot 4^n + \left(1 - \frac{1}{2^n}\right) \cdot 4^n \\&= 7 \cdot 4^n - 2^n.\end{aligned}$$



Examples

- Solve $T(n) = 3T(n-1) + n$ with $T(0) = 10$



Examples

- Solve $T(n) = 3T(n-1) + n$ with $T(0) = 10$

$$\begin{aligned} T(n) &= 10 \cdot 3^n + \sum_{i=1}^n 3^{n-i} \cdot i \\ &= 10 \cdot 3^n + 3^n \sum_{i=1}^n i \cdot 3^{-i} \end{aligned}$$



Examples

- Solve $T(n) = 3T(n-1) + n$ with $T(0) = 10$

$$\begin{aligned} T(n) &= 10 \cdot 3^n + \sum_{i=1}^n 3^{n-i} \cdot i \\ &= 10 \cdot 3^n + 3^n \sum_{i=1}^n i \cdot 3^{-i} \end{aligned}$$

Theorem. For any real number $x \neq 1$,

$$\sum_{i=1}^n ix^i = \frac{nx^{n+2} - (n+1)x^{n+1} + x}{(1-x)^2}.$$



Examples

- Solve $T(n) = 3T(n-1) + n$ with $T(0) = 10$

$$\begin{aligned}T(n) &= 10 \cdot 3^n + \sum_{i=1}^n 3^{n-i} \cdot i \\&= 10 \cdot 3^n + 3^n \sum_{i=1}^n i \cdot 3^{-i} \\&= 10 \cdot 3^n + 3^n \left(-\frac{3}{2}(n+1)3^{-(n+1)} - \frac{3}{4}3^{-(n+1)} + \frac{3}{4} \right) \\&= \frac{43}{4}3^n - \frac{n+1}{2} - \frac{1}{4}.\end{aligned}$$



Growth Rates of Solutions to Recurrences

- Divide and conquer algorithms
- Iterating recurrences
- Three different behaviors



Divide and conquer algorithms

- We just analyzed recurrences of the form

$$T(n) = \begin{cases} b & \text{if } n = 0 \\ r \cdot T(n-1) + a & \text{if } n > 0 \end{cases}$$



Divide and conquer algorithms

- We just analyzed recurrences of the form

$$T(n) = \begin{cases} b & \text{if } n = 0 \\ r \cdot T(n-1) + a & \text{if } n > 0 \end{cases}$$

These corresponded to the analysis of recursive algorithms in which a problem of size n is solved by recursively solving a problem of size $n - 1$.



Divide and conquer algorithms

- We just analyzed recurrences of the form

$$T(n) = \begin{cases} b & \text{if } n = 0 \\ r \cdot T(n-1) + a & \text{if } n > 0 \end{cases}$$

These corresponded to the analysis of recursive algorithms in which a problem of size n is solved by recursively solving a problem of size $n - 1$.

$$T(n) = r^n b + a \frac{1 - r^n}{1 - r}$$



Divide and conquer algorithms

- We just analyzed recurrences of the form

$$T(n) = \begin{cases} b & \text{if } n = 0 \\ r \cdot T(n-1) + a & \text{if } n > 0 \end{cases}$$

These corresponded to the analysis of recursive algorithms in which a problem of size n is solved by recursively solving a problem of size $n-1$.

$$T(n) = r^n b + a \frac{1 - r^n}{1 - r}$$

We will now look at recurrences of the form

$$T(n) = \begin{cases} \text{something given} & \text{if } n \leq n_0 \\ r \cdot T(n/m) + a & \text{if } n > n_0 \end{cases}$$



Binary Search

- Someone has chosen a number x between 1 and n . We need to discover x .



Binary Search

- Someone has chosen a number x between 1 and n . We need to discover x .

We are only allowed to ask **two types of questions**:



Binary Search

- Someone has chosen a number x between 1 and n . We need to discover x .

We are only allowed to ask two types of questions:

- ◇ Is x greater than k ?
- ◇ Is x equal to k ?



Binary Search

- Someone has chosen a number x between 1 and n . We need to discover x .

We are only allowed to ask two types of questions:

- ◇ Is x greater than k ?
- ◇ Is x equal to k ?

Our strategy will be to always ask greater than questions, at each step halving our search range, until the range only contains one number, when we ask a final equal to question.



Binary Search Example

1

32

48

64



Binary Search Example

1

32

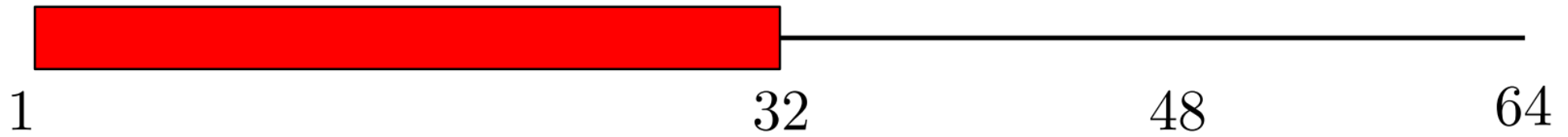
48

64

Is $x > 32$?



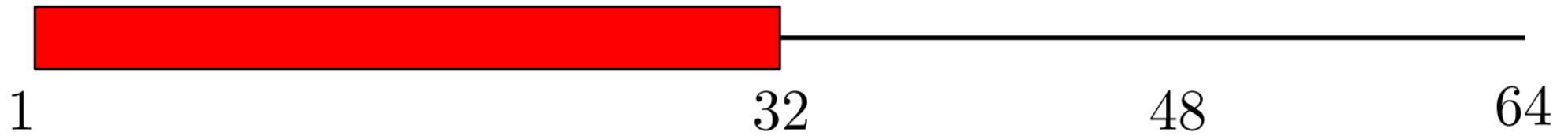
Binary Search Example



Is $x > 32$? Answer: Yes



Binary Search Example

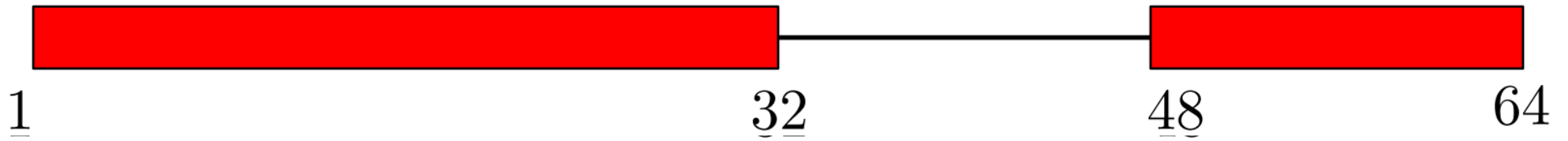


Is $x > 32$? Answer: Yes

Is $x > 48$?



Binary Search Example

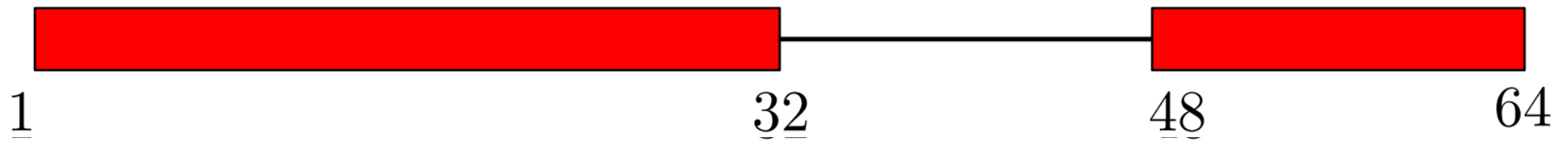


Is $x > 32$? Answer: Yes

Is $x > 48$? Answer: No



Binary Search Example



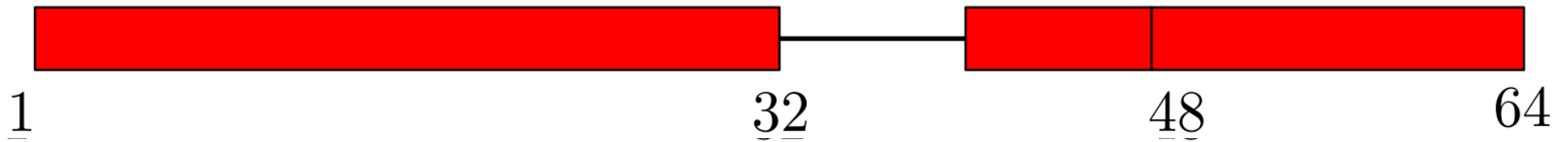
Is $x > 32$? Answer: Yes

Is $x > 48$? Answer: No

Is $x > 40$?



Binary Search Example



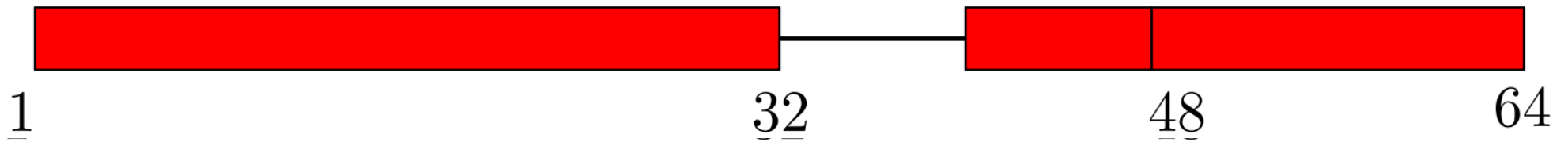
Is $x > 32$? Answer: Yes

Is $x > 48$? Answer: No

Is $x > 40$? Answer: No



Binary Search Example



Is $x > 32$? Answer: Yes

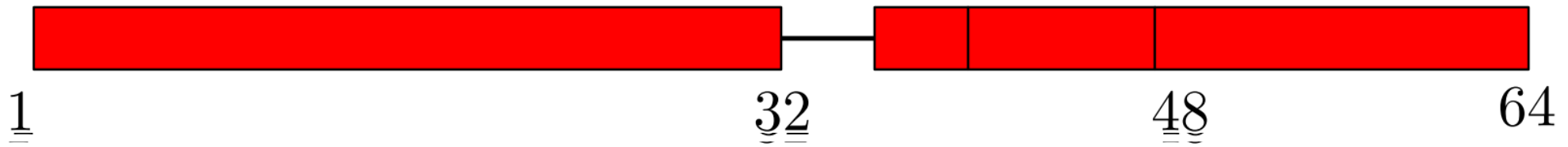
Is $x > 48$? Answer: No

Is $x > 40$? Answer: No

Is $x > 36$?



Binary Search Example



Is $x > 32$? Answer: Yes

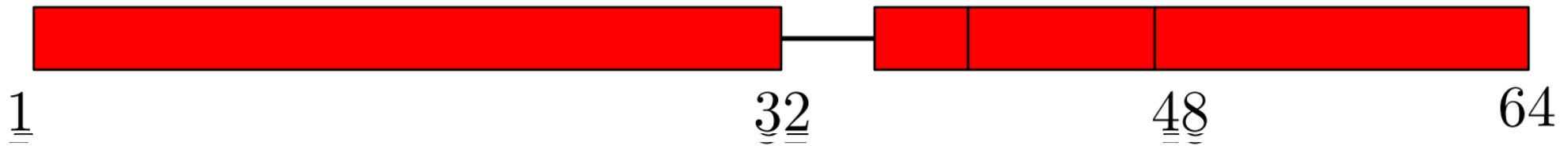
Is $x > 48$? Answer: No

Is $x > 40$? Answer: No

Is $x > 36$? Answer: No



Binary Search Example



Is $x > 32$? Answer: Yes

Is $x > 48$? Answer: No

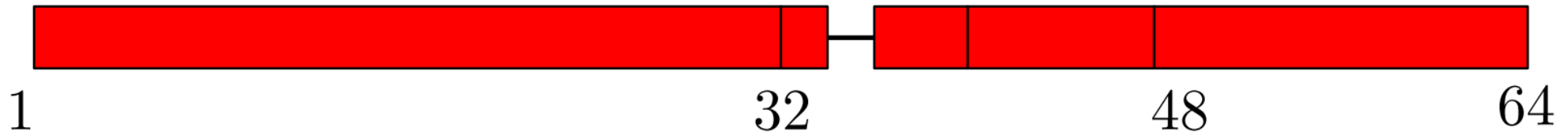
Is $x > 40$? Answer: No

Is $x > 36$? Answer: No

Is $x > 34$?



Binary Search Example



Is $x > 32$? Answer: Yes

Is $x > 48$? Answer: No

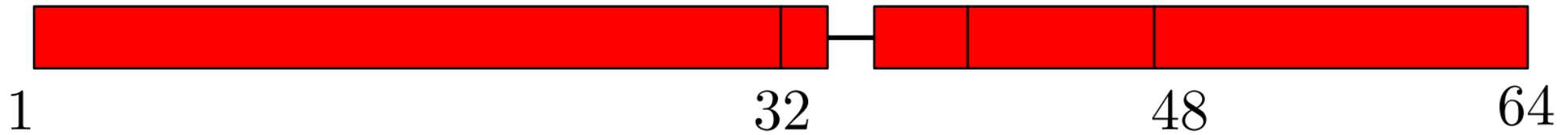
Is $x > 40$? Answer: No

Is $x > 36$? Answer: No

Is $x > 34$? Answer: Yes



Binary Search Example



Is $x > 32$? Answer: Yes

Is $x > 48$? Answer: No

Is $x > 40$? Answer: No

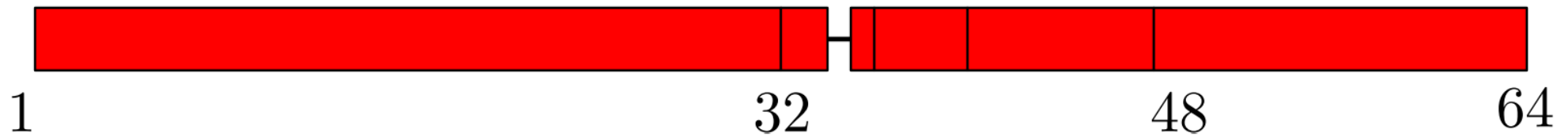
Is $x > 36$? Answer: No

Is $x > 34$? Answer: Yes

Is $x > 35$?



Binary Search Example



Is $x > 32$? Answer: Yes

Is $x > 48$? Answer: No

Is $x > 40$? Answer: No

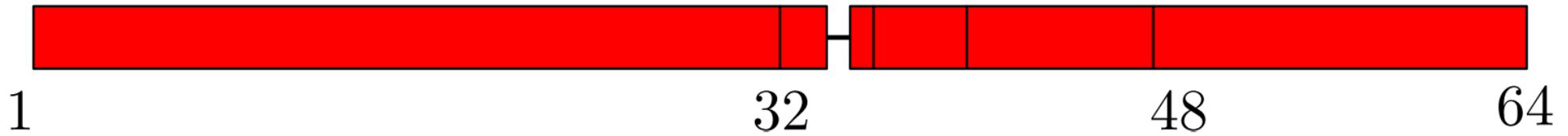
Is $x > 36$? Answer: No

Is $x > 34$? Answer: Yes

Is $x > 35$? Answer: No



Binary Search Example



Is $x > 32$? Answer: Yes

Is $x > 48$? Answer: No

Is $x > 40$? Answer: No

Is $x > 36$? Answer: No

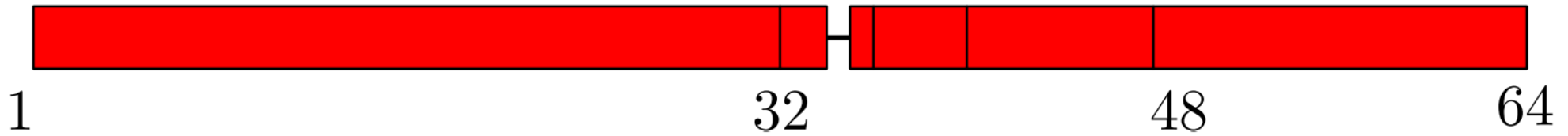
Is $x > 34$? Answer: Yes

Is $x > 35$? Answer: No

Is $x = 35$?



Binary Search Example



Is $x > 32$?	Answer: Yes
Is $x > 48$?	Answer: No
Is $x > 40$?	Answer: No
Is $x > 36$?	Answer: No
Is $x > 34$?	Answer: Yes
Is $x > 35$?	Answer: No
Is $x = 35$?	Answer: BINGO!



Binary Search Example

- **Method**: Each guess **reduces** the problem to one in which the range is only **half** as big.



Binary Search Example

- **Method**: Each guess **reduces** the problem to one in which the range is only **half** as big.

This **divides** the original problem into one that is only half as big; we can now (**recursively**) **conquer** this smaller problem.



Binary Search Example

- **Method:** Each guess **reduces** the problem to one in which the range is only **half** as big.

This **divides** the original problem into one that is only half as big; we can now (**recursively**) **conquer** this smaller problem.

Note: When n is a power of 2, $T(n)$, the number of questions in a binary search on $[1, n]$, satisfies

$$T(n) = \begin{cases} 1 & \text{if } n = 1 \\ T(n/2) + 1 & \text{if } n \geq 2 \end{cases}$$



Binary Search Example

- **Method:** Each guess **reduces** the problem to one in which the range is only **half** as big.

This **divides** the original problem into one that is only half as big; we can now (**recursively**) **conquer** this smaller problem.

Note: When n is a power of 2, $T(n)$, the number of questions in a binary search on $[1, n]$, satisfies

$$T(n) = \begin{cases} 1 & \text{if } n = 1 \\ T(n/2) + 1 & \text{if } n \geq 2 \end{cases}$$

This can also be proved **inductively**, similar to the tower of Hanoi recurrence.



Binary Search Example

- $T(n)$: number of questions in a binary search on $[1, n]$



Binary Search Example

- $T(n)$: number of questions in a binary search on $[1, n]$

Assume: n is a power of 2. Give recurrence for $T(n)$



Binary Search Example

- $T(n)$: number of questions in a binary search on $[1, n]$

Assume: n is a power of 2. Give recurrence for $T(n)$

$$T(n) = \begin{cases} 1 & \text{if } n = 1 \\ T(n/2) + 1 & \text{if } n \geq 2 \end{cases}$$



Binary Search Example

- $T(n)$: number of questions in a binary search on $[1, n]$

Assume: n is a power of 2. Give recurrence for $T(n)$

$$T(n) = \begin{cases} 1 & \text{if } n = 1 \\ T(n/2) + 1 & \text{if } n \geq 2 \end{cases}$$

Number of questions needed for **binary search** on n items is:



Binary Search Example

- $T(n)$: number of questions in a binary search on $[1, n]$

Assume: n is a power of 2. Give recurrence for $T(n)$

$$T(n) = \begin{cases} 1 & \text{if } n = 1 \\ T(n/2) + 1 & \text{if } n \geq 2 \end{cases}$$

Number of questions needed for **binary search** on n items is:

first step

+

time to perform binary search on the remaining $n/2$ items



Binary Search Example

- $T(n)$: number of questions in a binary search on $[1, n]$

Assume: n is a power of 2. Give recurrence for $T(n)$

$$T(n) = \begin{cases} 1 & \text{if } n = 1 \\ T(n/2) + 1 & \text{if } n \geq 2 \end{cases}$$

Number of questions needed for **binary search** on n items is:

first step

+

time to perform binary search on the remaining $n/2$ items

Base case (1 item): $T(1) = 1$ to ask: “**Is the number k ?**”



Binary Search Example

$$(*) \quad T(n) = \begin{cases} C_1 & \text{if } n = 1 \\ T(\lceil n/2 \rceil) + C_2 & \text{if } n \geq 2 \end{cases}$$

For simplicity, we will (usually) assume that n is a power of 2 (or sometimes 3 or 4) and also often that constants such as C_1, C_2 are 1. This will let us replace a recurrence such as (*) by one such as (**).



Binary Search Example

$$(*) \quad T(n) = \begin{cases} C_1 & \text{if } n = 1 \\ T(\lceil n/2 \rceil) + C_2 & \text{if } n \geq 2 \end{cases}$$

For simplicity, we will (usually) assume that n is a power of 2 (or sometimes 3 or 4) and also often that constants such as C_1, C_2 are 1. This will let us replace a recurrence such as (*) by one such as (**).

$$(**) \quad T(n) = \begin{cases} 1 & \text{if } n = 1 \\ T(n/2) + 1 & \text{if } n \geq 2 \end{cases}$$



Binary Search Example

$$(*) \quad T(n) = \begin{cases} C_1 & \text{if } n = 1 \\ T(\lceil n/2 \rceil) + C_2 & \text{if } n \geq 2 \end{cases}$$

For simplicity, we will (usually) assume that n is a power of 2 (or sometimes 3 or 4) and also often that constants such as C_1, C_2 are 1. This will let us replace a recurrence such as (*) by one such as (**).

$$(**) \quad T(n) = \begin{cases} 1 & \text{if } n = 1 \\ T(n/2) + 1 & \text{if } n \geq 2 \end{cases}$$

In practice, the solution of (*) will be very close to that of (**) (this can be proved mathematically). Hence, we can restrict attention to (**).



Growth Rates of Solutions to Recurrences

- Divide and conquer algorithms
- Iterating recurrences
- Three different behaviors



Iterating Recurrences: Example 1

■

$$(*) \quad T(n) = \begin{cases} T(1) & \text{if } n = 1 \\ 2T(n/2) + n & \text{if } n \geq 2 \end{cases}$$



Iterating Recurrences: Example 1

■

$$(*) \quad T(n) = \begin{cases} T(1) & \text{if } n = 1 \\ 2T(n/2) + n & \text{if } n \geq 2 \end{cases}$$

This corresponds to solving a problem of size n , by

- (i) solving 2 subproblems of size $n/2$ and
- (ii) doing n units of additional work

or using $T(1)$ work for “bottom” case of $n = 1$



Iterating Recurrences: Example 1

■

$$(*) \quad T(n) = \begin{cases} T(1) & \text{if } n = 1 \\ 2T(n/2) + n & \text{if } n \geq 2 \end{cases}$$

This corresponds to solving a problem of size n , by

- (i) solving 2 subproblems of size $n/2$ and
- (ii) doing n units of additional work

or using $T(1)$ work for “bottom” case of $n = 1$

In the course “Analysis of Algorithms”, this is exactly how **Mergesort** works.



Iterating Recurrences: Example 1

$$(*) \quad T(n) = \begin{cases} T(1) & \text{if } n = 1 \\ 2T(n/2) + n & \text{if } n \geq 2 \end{cases}$$

This corresponds to solving a problem of size n , by

- (i) solving 2 subproblems of size $n/2$ and
- (ii) doing n units of additional work

or using $T(1)$ work for “bottom” case of $n = 1$

In the course “Analysis of Algorithms”, this is exactly how **Mergesort** works.

We now see how to solve $(*)$ by algebraically iterating the recurrence.



Iterating Recurrences: Example 1

- Algebraically iterating the recurrence

$$T(n) = 2T\left(\frac{n}{2}\right) + n$$



Iterating Recurrences: Example 1

- Algebraically iterating the recurrence

Assume that n is a power of 2

$$T(n) = 2T\left(\frac{n}{2}\right) + n = 2\left(2T\left(\frac{n}{4}\right) + \frac{n}{2}\right) + n$$



Iterating Recurrences: Example 1

- Algebraically iterating the recurrence

Assume that n is a power of 2

$$\begin{aligned} T(n) &= 2T\left(\frac{n}{2}\right) + n &= 2\left(2T\left(\frac{n}{4}\right) + \frac{n}{2}\right) + n \\ &= 4T\left(\frac{n}{4}\right) + 2n &= 4\left(2T\left(\frac{n}{8}\right) + \frac{n}{4}\right) + 2n \end{aligned}$$



Iterating Recurrences: Example 1

- Algebraically iterating the recurrence

Assume that n is a power of 2

$$\begin{aligned}T(n) &= 2T\left(\frac{n}{2}\right) + n &= 2\left(2T\left(\frac{n}{4}\right) + \frac{n}{2}\right) + n \\&= 4T\left(\frac{n}{4}\right) + 2n &= 4\left(2T\left(\frac{n}{8}\right) + \frac{n}{4}\right) + 2n \\&= 8T\left(\frac{n}{8}\right) + 3n\end{aligned}$$



Iterating Recurrences: Example 1

- Algebraically iterating the recurrence

Assume that n is a power of 2

$$\begin{aligned}T(n) &= 2T\left(\frac{n}{2}\right) + n &= 2\left(2T\left(\frac{n}{4}\right) + \frac{n}{2}\right) + n \\&= 4T\left(\frac{n}{4}\right) + 2n &= 4\left(2T\left(\frac{n}{8}\right) + \frac{n}{4}\right) + 2n \\&= 8T\left(\frac{n}{8}\right) + 3n \\&\quad \vdots \quad \vdots \\&= 2^i T\left(\frac{n}{2^i}\right) + in\end{aligned}$$



Iterating Recurrences: Example 1

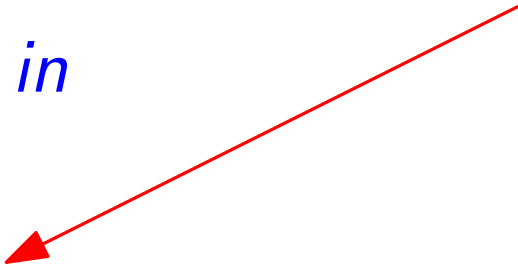
- Algebraically iterating the recurrence

Assume that n is a power of 2

$$\begin{aligned}T(n) &= 2T\left(\frac{n}{2}\right) + n &= 2\left(2T\left(\frac{n}{4}\right) + \frac{n}{2}\right) + n \\&= 4T\left(\frac{n}{4}\right) + 2n &= 4\left(2T\left(\frac{n}{8}\right) + \frac{n}{4}\right) + 2n \\&= 8T\left(\frac{n}{8}\right) + 3n\end{aligned}$$

$$\begin{aligned}&\vdots \quad \vdots \\&= 2^i T\left(\frac{n}{2^i}\right) + in \\&\vdots \quad \vdots \\&= 2^{\log_2 n} T\left(\frac{n}{2^{\log_2 n}}\right) + (\log_2 n)n\end{aligned}$$

End when $i = \log_2 n$





Iterating Recurrences: Example 1

- Algebraically iterating the recurrence

Assume that n is a power of 2

$$T(n) = 2T\left(\frac{n}{2}\right) + n = 2\left(2T\left(\frac{n}{4}\right) + \frac{n}{2}\right) + n$$

$$= 4T\left(\frac{n}{4}\right) + 2n = 4\left(2T\left(\frac{n}{8}\right) + \frac{n}{4}\right) + 2n$$

$$= 8T\left(\frac{n}{8}\right) + 3n$$

$$\vdots \quad \vdots$$
$$= 2^i T\left(\frac{n}{2^i}\right) + in$$

End when $i = \log_2 n$

$$\vdots \quad \vdots$$
$$= 2^{\log_2 n} T\left(\frac{n}{2^{\log_2 n}}\right) + (\log_2 n)n$$

$$= nT(1) + n\log_2 n$$



Iterating Recurrences: Example 1

- We just iterated the recurrence to derive that the solution to

$$(*) \quad T(n) = \begin{cases} T(1) & \text{if } n = 1 \\ 2T(n/2) + n & \text{if } n \geq 2 \end{cases}$$

is $nT(1) + n \log_2 n$.



Iterating Recurrences: Example 1

- We just iterated the recurrence to derive that the solution to

$$(*) \quad T(n) = \begin{cases} T(1) & \text{if } n = 1 \\ 2T(n/2) + n & \text{if } n \geq 2 \end{cases}$$

is $nT(1) + n \log_2 n$.

Note: Technically, we still need to use **induction** to prove that our solution is correct. Practically, we **never** explicitly perform this step, since it is obvious how the induction would work.



Iterating Recurrences: Example 2

■

$$(*) \quad T(n) = \begin{cases} 1 & \text{if } n = 1 \\ T(n/2) + 1 & \text{if } n \geq 2 \end{cases}$$



Iterating Recurrences: Example 2

■

$$(*) \quad T(n) = \begin{cases} 1 & \text{if } n = 1 \\ T(n/2) + 1 & \text{if } n \geq 2 \end{cases}$$

$$T(n) = T\left(\frac{n}{2}\right) + 1$$



Iterating Recurrences: Example 2



$$(*) \quad T(n) = \begin{cases} 1 & \text{if } n = 1 \\ T(n/2) + 1 & \text{if } n \geq 2 \end{cases}$$

$$T(n) = T\left(\frac{n}{2}\right) + 1 = \left(T\left(\frac{n}{2^2}\right) + 1\right) + 1$$



Iterating Recurrences: Example 2



$$(*) \quad T(n) = \begin{cases} 1 & \text{if } n = 1 \\ T(n/2) + 1 & \text{if } n \geq 2 \end{cases}$$

$$\begin{aligned} T(n) &= T\left(\frac{n}{2}\right) + 1 &= \left(T\left(\frac{n}{2^2}\right) + 1\right) + 1 \\ &= T\left(\frac{n}{2^2}\right) + 2 \end{aligned}$$



Iterating Recurrences: Example 2

■

$$(*) \quad T(n) = \begin{cases} 1 & \text{if } n = 1 \\ T(n/2) + 1 & \text{if } n \geq 2 \end{cases}$$

$$\begin{aligned} T(n) &= T\left(\frac{n}{2}\right) + 1 &&= \left(T\left(\frac{n}{2^2}\right) + 1\right) + 1 \\ &= T\left(\frac{n}{2^2}\right) + 2 &&= \left(T\left(\frac{n}{2^3}\right) + 1\right) + 2 \end{aligned}$$



Iterating Recurrences: Example 2

■

$$(*) \quad T(n) = \begin{cases} 1 & \text{if } n = 1 \\ T(n/2) + 1 & \text{if } n \geq 2 \end{cases}$$

$$\begin{aligned} T(n) &= T\left(\frac{n}{2}\right) + 1 &&= \left(T\left(\frac{n}{2^2}\right) + 1\right) + 1 \\ &= T\left(\frac{n}{2^2}\right) + 2 &&= \left(T\left(\frac{n}{2^3}\right) + 1\right) + 2 \\ &= T\left(\frac{n}{2^3}\right) + 3 \end{aligned}$$



Iterating Recurrences: Example 2



$$(*) \quad T(n) = \begin{cases} 1 & \text{if } n = 1 \\ T(n/2) + 1 & \text{if } n \geq 2 \end{cases}$$

$$\begin{aligned} T(n) &= T\left(\frac{n}{2}\right) + 1 &&= \left(T\left(\frac{n}{2^2}\right) + 1\right) + 1 \\ &= T\left(\frac{n}{2^2}\right) + 2 &&= \left(T\left(\frac{n}{2^3}\right) + 1\right) + 2 \\ &= T\left(\frac{n}{2^3}\right) + 3 \\ &\quad \vdots \quad \quad \quad \vdots \\ &= T\left(\frac{n}{2^i}\right) + i \end{aligned}$$



Iterating Recurrences: Example 2



$$(*) \quad T(n) = \begin{cases} 1 & \text{if } n = 1 \\ T(n/2) + 1 & \text{if } n \geq 2 \end{cases}$$

$$T(n) = T\left(\frac{n}{2}\right) + 1 = \left(T\left(\frac{n}{2^2}\right) + 1\right) + 1$$

$$= T\left(\frac{n}{2^2}\right) + 2 = \left(T\left(\frac{n}{2^3}\right) + 1\right) + 2$$

$$= T\left(\frac{n}{2^3}\right) + 3$$

$$\begin{array}{c} \vdots \\ \vdots \\ = T\left(\frac{n}{2^i}\right) + i \end{array}$$

$$\begin{array}{c} \vdots \\ \vdots \\ = T\left(\frac{n}{2^{\log_2 n}}\right) + \log_2 n \end{array}$$



Iterating Recurrences: Example 2

■

$$(*) \quad T(n) = \begin{cases} 1 & \text{if } n = 1 \\ T(n/2) + 1 & \text{if } n \geq 2 \end{cases}$$

$$T(n) = T\left(\frac{n}{2}\right) + 1 = \left(T\left(\frac{n}{2^2}\right) + 1\right) + 1$$

$$= T\left(\frac{n}{2^2}\right) + 2 = \left(T\left(\frac{n}{2^3}\right) + 1\right) + 2$$

$$= T\left(\frac{n}{2^3}\right) + 3$$

$$\begin{array}{c} \vdots \\ \vdots \\ = T\left(\frac{n}{2^i}\right) + i \end{array}$$

$$\begin{array}{c} \vdots \\ \vdots \\ = T\left(\frac{n}{2^{\log_2 n}}\right) + \log_2 n = 1 + \log_2 n \end{array}$$



Iterating Recurrences: Example 3

■

$$(*) \quad T(n) = \begin{cases} 1 & \text{if } n = 1 \\ T(n/2) + n & \text{if } n \geq 2 \end{cases}$$



Iterating Recurrences: Example 3

■

$$(*) \quad T(n) = \begin{cases} 1 & \text{if } n = 1 \\ T(n/2) + n & \text{if } n \geq 2 \end{cases}$$

$$\begin{aligned} T(n) &= T\left(\frac{n}{2}\right) + n \\ &= T\left(\frac{n}{2^2}\right) + \frac{n}{2} + n \end{aligned}$$



Iterating Recurrences: Example 3

■

$$(*) \quad T(n) = \begin{cases} 1 & \text{if } n = 1 \\ T(n/2) + n & \text{if } n \geq 2 \end{cases}$$

$$\begin{aligned} T(n) &= T\left(\frac{n}{2}\right) + n \\ &= T\left(\frac{n}{2^2}\right) + \frac{n}{2} + n \\ &= T\left(\frac{n}{2^3}\right) + \frac{n}{2^2} + \frac{n}{2} + n \end{aligned}$$



Iterating Recurrences: Example 3

■

$$(*) \quad T(n) = \begin{cases} 1 & \text{if } n = 1 \\ T(n/2) + n & \text{if } n \geq 2 \end{cases}$$

$$\begin{aligned} T(n) &= T\left(\frac{n}{2}\right) + n \\ &= T\left(\frac{n}{2^2}\right) + \frac{n}{2} + n \\ &= T\left(\frac{n}{2^3}\right) + \frac{n}{2^2} + \frac{n}{2} + n \\ &\quad \vdots \\ &= T\left(\frac{n}{2^i}\right) + \frac{n}{2^{i-1}} + \cdots + \frac{n}{2^2} + \frac{n}{2} + n \end{aligned}$$



Iterating Recurrences: Example 3

$$(*) \quad T(n) = \begin{cases} 1 & \text{if } n = 1 \\ T(n/2) + n & \text{if } n \geq 2 \end{cases}$$

$$T(n) = T\left(\frac{n}{2}\right) + n$$

$$= T\left(\frac{n}{2^2}\right) + \frac{n}{2} + n$$

$$= T\left(\frac{n}{2^3}\right) + \frac{n}{2^2} + \frac{n}{2} + n$$

$$\vdots \quad \vdots$$

$$= T\left(\frac{n}{2^i}\right) + \frac{n}{2^{i-1}} + \cdots + \frac{n}{2^2} + \frac{n}{2} + n$$

$$\vdots \quad \vdots$$

$$= T\left(\frac{n}{2^{\log_2 n}}\right) + \frac{n}{2^{\log_2 n - 1}} + \cdots + \frac{n}{2^2} + \frac{n}{2} + n$$



Iterating Recurrences: Example 3

$$(*) \quad T(n) = \begin{cases} 1 & \text{if } n = 1 \\ T(n/2) + n & \text{if } n \geq 2 \end{cases}$$

$$T(n) = T\left(\frac{n}{2}\right) + n$$

$$= T\left(\frac{n}{2^2}\right) + \frac{n}{2} + n$$

$$= T\left(\frac{n}{2^3}\right) + \frac{n}{2^2} + \frac{n}{2} + n$$

$$\vdots \quad \vdots$$

$$= T\left(\frac{n}{2^i}\right) + \frac{n}{2^{i-1}} + \cdots + \frac{n}{2^2} + \frac{n}{2} + n$$

$$\vdots \quad \vdots$$

$$= T\left(\frac{n}{2^{\log_2 n}}\right) + \frac{n}{2^{\log_2 n - 1}} + \cdots + \frac{n}{2^2} + \frac{n}{2} + n$$

$$= 1 + 2 + 2^2 + \cdots + \frac{n}{2^2} + \frac{n}{2} + n$$



Iterating Recurrences: Example 3

$$(*) \quad T(n) = \begin{cases} 1 & \text{if } n = 1 \\ T(n/2) + n & \text{if } n \geq 2 \end{cases}$$

$$T(n) = T\left(\frac{n}{2}\right) + n$$

$$= T\left(\frac{n}{2^2}\right) + \frac{n}{2} + n$$

$$= T\left(\frac{n}{2^3}\right) + \frac{n}{2^2} + \frac{n}{2} + n$$

$$\vdots \quad \vdots$$

$$= T\left(\frac{n}{2^i}\right) + \frac{n}{2^{i-1}} + \cdots + \frac{n}{2^2} + \frac{n}{2} + n$$

$$\vdots \quad \vdots$$

$$= T\left(\frac{n}{2^{\log_2 n}}\right) + \frac{n}{2^{\log_2 n - 1}} + \cdots + \frac{n}{2^2} + \frac{n}{2} + n$$

$$= 1 + 2 + 2^2 + \cdots + \frac{n}{2^2} + \frac{n}{2} + n = \Theta(n)$$



Iterating Recurrences: Example 4

■

$$(*) \quad T(n) = \begin{cases} 1 & \text{if } n < 3 \\ 3T(n/3) + n & \text{if } n \geq 3 \end{cases}$$



Iterating Recurrences: Example 4

■

$$(*) \quad T(n) = \begin{cases} 1 & \text{if } n < 3 \\ 3T(n/3) + n & \text{if } n \geq 3 \end{cases}$$

$$T(n) = 3T\left(\frac{n}{3}\right) + n$$



Iterating Recurrences: Example 4

■

$$(*) \quad T(n) = \begin{cases} 1 & \text{if } n < 3 \\ 3T(n/3) + n & \text{if } n \geq 3 \end{cases}$$

$$T(n) = 3T\left(\frac{n}{3}\right) + n = 3\left(3T\left(\frac{n}{3^2}\right) + \frac{n}{3}\right) + n$$



Iterating Recurrences: Example 4

■

$$(*) \quad T(n) = \begin{cases} 1 & \text{if } n < 3 \\ 3T(n/3) + n & \text{if } n \geq 3 \end{cases}$$

$$\begin{aligned} T(n) &= 3T\left(\frac{n}{3}\right) + n &= 3\left(3T\left(\frac{n}{3^2}\right) + \frac{n}{3}\right) + n \\ &= 3^2 T\left(\frac{n}{3^2}\right) + 2n \end{aligned}$$



Iterating Recurrences: Example 4

■

$$(*) \quad T(n) = \begin{cases} 1 & \text{if } n < 3 \\ 3T(n/3) + n & \text{if } n \geq 3 \end{cases}$$

$$\begin{aligned} T(n) &= 3T\left(\frac{n}{3}\right) + n &= 3\left(3T\left(\frac{n}{3^2}\right) + \frac{n}{3}\right) + n \\ &= 3^2 T\left(\frac{n}{3^2}\right) + 2n &= 3^2\left(3T\left(\frac{n}{3^3}\right) + \frac{n}{3^2}\right) + 2n \end{aligned}$$



Iterating Recurrences: Example 4

■

$$(*) \quad T(n) = \begin{cases} 1 & \text{if } n < 3 \\ 3T(n/3) + n & \text{if } n \geq 3 \end{cases}$$

$$\begin{aligned} T(n) &= 3T\left(\frac{n}{3}\right) + n &= 3\left(3T\left(\frac{n}{3^2}\right) + \frac{n}{3}\right) + n \\ &= 3^2 T\left(\frac{n}{3^2}\right) + 2n &= 3^2\left(3T\left(\frac{n}{3^3}\right) + \frac{n}{3^2}\right) + 2n \\ &= 3^3 T\left(\frac{n}{3^3}\right) + 3n \end{aligned}$$



Iterating Recurrences: Example 4

■

$$(*) \quad T(n) = \begin{cases} 1 & \text{if } n < 3 \\ 3T(n/3) + n & \text{if } n \geq 3 \end{cases}$$

$$\begin{aligned} T(n) &= 3T\left(\frac{n}{3}\right) + n &= 3\left(3T\left(\frac{n}{3^2}\right) + \frac{n}{3}\right) + n \\ &= 3^2 T\left(\frac{n}{3^2}\right) + 2n &= 3^2\left(3T\left(\frac{n}{3^3}\right) + \frac{n}{3^2}\right) + 2n \\ &= 3^3 T\left(\frac{n}{3^3}\right) + 3n \\ &\quad \vdots \quad \vdots \\ &= 3^i T\left(\frac{n}{3^i}\right) + in \end{aligned}$$



Iterating Recurrences: Example 4

$$(*) \quad T(n) = \begin{cases} 1 & \text{if } n < 3 \\ 3T(n/3) + n & \text{if } n \geq 3 \end{cases}$$

$$\begin{aligned} T(n) &= 3T\left(\frac{n}{3}\right) + n &= 3\left(3T\left(\frac{n}{3^2}\right) + \frac{n}{3}\right) + n \\ &= 3^2 T\left(\frac{n}{3^2}\right) + 2n &= 3^2\left(3T\left(\frac{n}{3^3}\right) + \frac{n}{3^2}\right) + 2n \\ &= 3^3 T\left(\frac{n}{3^3}\right) + 3n \\ &\quad \vdots \quad \vdots \\ &= 3^i T\left(\frac{n}{3^i}\right) + in \\ &\quad \vdots \quad \vdots \\ &= 3^{\log_3 n} T\left(\frac{n}{3^{\log_3 n}}\right) + n \log_3 n \end{aligned}$$



Iterating Recurrences: Example 4

$$(*) \quad T(n) = \begin{cases} 1 & \text{if } n < 3 \\ 3T(n/3) + n & \text{if } n \geq 3 \end{cases}$$

$$\begin{aligned} T(n) &= 3T\left(\frac{n}{3}\right) + n &= 3\left(3T\left(\frac{n}{3^2}\right) + \frac{n}{3}\right) + n \\ &= 3^2 T\left(\frac{n}{3^2}\right) + 2n &= 3^2\left(3T\left(\frac{n}{3^3}\right) + \frac{n}{3^2}\right) + 2n \\ &= 3^3 T\left(\frac{n}{3^3}\right) + 3n \\ &\quad \vdots \quad \vdots \\ &= 3^i T\left(\frac{n}{3^i}\right) + in \\ &\quad \vdots \quad \vdots \\ &= 3^{\log_3 n} T\left(\frac{n}{3^{\log_3 n}}\right) + n \log_3 n = n + n \log_3 n \end{aligned}$$



Iterating Recurrences: Example 5

■

$$(*) \quad T(n) = \begin{cases} 1 & \text{if } n = 1 \\ 4T(n/2) + n & \text{if } n \geq 2 \end{cases}$$



Iterating Recurrences: Example 5

■

$$(*) \quad T(n) = \begin{cases} 1 & \text{if } n = 1 \\ 4T(n/2) + n & \text{if } n \geq 2 \end{cases}$$

$$T(n) = 4T\left(\frac{n}{2}\right) + n$$



Iterating Recurrences: Example 5

■

$$(*) \quad T(n) = \begin{cases} 1 & \text{if } n = 1 \\ 4T(n/2) + n & \text{if } n \geq 2 \end{cases}$$

$$T(n) = 4T\left(\frac{n}{2}\right) + n = 4\left(4T\left(\frac{n}{2^2}\right) + \frac{n}{2}\right) + n$$



Iterating Recurrences: Example 5

■

$$(*) \quad T(n) = \begin{cases} 1 & \text{if } n = 1 \\ 4T(n/2) + n & \text{if } n \geq 2 \end{cases}$$

$$\begin{aligned} T(n) &= 4T\left(\frac{n}{2}\right) + n &= 4\left(4T\left(\frac{n}{2^2}\right) + \frac{n}{2}\right) + n \\ &= 4^2 T\left(\frac{n}{2^2}\right) + \frac{4}{2}n + n \end{aligned}$$



Iterating Recurrences: Example 5

■

$$(*) \quad T(n) = \begin{cases} 1 & \text{if } n = 1 \\ 4T(n/2) + n & \text{if } n \geq 2 \end{cases}$$

$$\begin{aligned} T(n) &= 4T\left(\frac{n}{2}\right) + n &= 4\left(4T\left(\frac{n}{2^2}\right) + \frac{n}{2}\right) + n \\ &= 4^2 T\left(\frac{n}{2^2}\right) + \frac{4}{2}n + n &= 4^2\left(4T\left(\frac{n}{2^3}\right) + \frac{n}{2^2}\right) + \frac{4}{2}n + n \end{aligned}$$



Iterating Recurrences: Example 5

■

$$(*) \quad T(n) = \begin{cases} 1 & \text{if } n = 1 \\ 4T(n/2) + n & \text{if } n \geq 2 \end{cases}$$

$$\begin{aligned} T(n) &= 4T\left(\frac{n}{2}\right) + n &&= 4\left(4T\left(\frac{n}{2^2}\right) + \frac{n}{2}\right) + n \\ &= 4^2 T\left(\frac{n}{2^2}\right) + \frac{4}{2}n + n &&= 4^2\left(4T\left(\frac{n}{2^3}\right) + \frac{n}{2^2}\right) + \frac{4}{2}n + n \\ &= 4^3 T\left(\frac{n}{2^3}\right) + \frac{4^2}{2^2}n + \frac{4}{2}n + n \end{aligned}$$



Iterating Recurrences: Example 5

■

$$(*) \quad T(n) = \begin{cases} 1 & \text{if } n = 1 \\ 4T(n/2) + n & \text{if } n \geq 2 \end{cases}$$

$$\begin{aligned} T(n) &= 4T\left(\frac{n}{2}\right) + n &&= 4\left(4T\left(\frac{n}{2^2}\right) + \frac{n}{2}\right) + n \\ &= 4^2 T\left(\frac{n}{2^2}\right) + \frac{4}{2}n + n &&= 4^2\left(4T\left(\frac{n}{2^3}\right) + \frac{n}{2^2}\right) + \frac{4}{2}n + n \\ &= 4^3 T\left(\frac{n}{2^3}\right) + \frac{4^2}{2^2}n + \frac{4}{2}n + n \\ &\quad \vdots \quad \quad \quad \vdots \\ &= 4^i T\left(\frac{n}{2^i}\right) + \frac{4^{i-1}}{2^{i-1}}n + \cdots + \frac{4^2}{2^2}n + n \end{aligned}$$



Iterating Recurrences: Example 5

■

$$(*) \quad T(n) = \begin{cases} 1 & \text{if } n = 1 \\ 4T(n/2) + n & \text{if } n \geq 2 \end{cases}$$

$$\begin{aligned} T(n) &= 4T\left(\frac{n}{2}\right) + n &&= 4\left(4T\left(\frac{n}{2^2}\right) + \frac{n}{2}\right) + n \\ &= 4^2 T\left(\frac{n}{2^2}\right) + \frac{4}{2}n + n &&= 4^2\left(4T\left(\frac{n}{2^3}\right) + \frac{n}{2^2}\right) + \frac{4}{2}n + n \\ &= 4^3 T\left(\frac{n}{2^3}\right) + \frac{4^2}{2^2}n + \frac{4}{2}n + n \\ &\quad \vdots \quad \vdots \\ &= 4^i T\left(\frac{n}{2^i}\right) + \frac{4^{i-1}}{2^{i-1}}n + \cdots + \frac{4^2}{2^2}n + n \\ &\quad \vdots \quad \vdots \\ &= 4^{\log_2 n} T\left(\frac{n}{2^{\log_2 n}}\right) + \frac{4^{\log_2 n - 1}}{2^{\log_2 n - 1}}n + \cdots + \frac{4}{2}n + n \end{aligned}$$



Iterating Recurrences: Example 5

■

$$(*) \quad T(n) = \begin{cases} 1 & \text{if } n = 1 \\ 4T(n/2) + n & \text{if } n \geq 2 \end{cases}$$

$$\begin{aligned} T(n) &= 4T\left(\frac{n}{2}\right) + n &&= 4\left(4T\left(\frac{n}{2^2}\right) + \frac{n}{2}\right) + n \\ &= 4^2 T\left(\frac{n}{2^2}\right) + \frac{4}{2}n + n &&= 4^2\left(4T\left(\frac{n}{2^3}\right) + \frac{n}{2^2}\right) + \frac{4}{2}n + n \\ &= 4^3 T\left(\frac{n}{2^3}\right) + \frac{4^2}{2^2}n + \frac{4}{2}n + n \\ &\quad \vdots \quad \vdots \\ &= 4^i T\left(\frac{n}{2^i}\right) + \frac{4^{i-1}}{2^{i-1}}n + \cdots + \frac{4^2}{2^2}n + n \\ &\quad \vdots \quad \vdots \\ &= 4^{\log_2 n} T\left(\frac{n}{2^{\log_2 n}}\right) + \frac{4^{\log_2 n - 1}}{2^{\log_2 n - 1}}n + \cdots + \frac{4}{2}n + n \\ &= 2n^2 - n \end{aligned}$$



Growth Rates of Solutions to Recurrences

- Divide and conquer algorithms
- Iteration recurrences
- Three different behaviors



Three Different Behaviors

- Compare the iteration for the recurrences

$$T(n) = 2T(n/2) + n$$

$$T(n) = T(n/2) + n$$

$$T(n) = 4T(n/2) + n$$



Three Different Behaviors

- Compare the iteration for the recurrences

$$T(n) = 2T(n/2) + n$$

$$T(n) = T(n/2) + n$$

$$T(n) = 4T(n/2) + n$$

- ◇ all three recurrences iterate $\log_2 n$ times
- ◇ in each case, size of subproblem in next iteration is **half** the size in the preceding iteration level



Three Different Behaviors

- **Theorem** Suppose that we have a recurrence of the form

$$T(n) = aT(n/2) + n,$$

where a is a positive integer and $T(1)$ is nonnegative. Then we have the following **big Θ** bounds on the solution:

1. If $a < 2$, then $T(n) = \Theta(n)$.
2. If $a = 2$, then $T(n) = \Theta(n \log n)$.
3. If $a > 2$, then $T(n) = \Theta(n^{\log_2 a})$.



Three Different Behaviors

- **Theorem** Suppose that we have a recurrence of the form

$$T(n) = aT(n/2) + n,$$

where a is a positive integer and $T(1)$ is nonnegative. Then we have the following **big Θ** bounds on the solution:

1. If $a < 2$, then $T(n) = \Theta(n)$.
2. If $a = 2$, then $T(n) = \Theta(n \log n)$.
3. If $a > 2$, then $T(n) = \Theta(n^{\log_2 a})$

Proof

We already proved Case 1 when $a = 1$ in Example 3.

(will not prove it for $1 < a < 2$)

We already proved Case 2 in Example 1.

We will now prove Case 3.



Iterating Recurrences

- $T(n) = aT(n/2) + n$, where $a > 2$. Assume that $n = 2^i$.



Iterating Recurrences

- $T(n) = aT(n/2) + n$, where $a > 2$. Assume that $n = 2^i$.

Iterating as in Example 5 gives

$$T(n) = a^i T\left(\frac{n}{2^i}\right) + \left(\frac{a^{i-1}}{2^{i-1}} + \frac{a^{i-2}}{2^{i-2}} + \cdots + \frac{a}{2} + 1\right) n$$



Iterating Recurrences

- $T(n) = aT(n/2) + n$, where $a > 2$. Assume that $n = 2^i$.

Iterating as in Example 5 gives

$$T(n) = a^i T\left(\frac{n}{2^i}\right) + \left(\frac{a^{i-1}}{2^{i-1}} + \frac{a^{i-2}}{2^{i-2}} + \cdots + \frac{a}{2} + 1\right) n$$

$$T(n) = a^{\log_2 n} T(1) + n \sum_{i=0}^{\log_2 n - 1} \left(\frac{a}{2}\right)^i$$

Work at
“bottom”

Iterated
Work



Total work

- The total work is

$$T(n) = a^{\log_2 n} T(1) + n \sum_{i=0}^{\log_2 n - 1} \left(\frac{a}{2}\right)^i$$



Total work

- The total work is

$$T(n) = a^{\log_2 n} T(1) + n \sum_{i=0}^{\log_2 n - 1} \left(\frac{a}{2}\right)^i$$

Since $a > 2$, the geometric series is Θ of the largest term.

$$n \sum_{i=0}^{\log_2 n - 1} \left(\frac{a}{2}\right)^i = n \frac{1 - (a/2)^{\log_2 n}}{1 - a/2} = n\Theta((a/2)^{\log_2 n - 1})$$



Total work

- n times the largest term in the geometric series is

$$n \left(\frac{a}{2}\right)^{\log_2 n - 1} = \frac{2}{a} \cdot \frac{n \cdot a^{\log_2 n}}{2^{\log_2 n}} = \frac{2}{a} \cdot \frac{n \cdot a^{\log_2 n}}{n} = \frac{2}{a} \cdot a^{\log_2 n}$$



Total work

- n times the largest term in the geometric series is

$$n \left(\frac{a}{2}\right)^{\log_2 n - 1} = \frac{2}{a} \cdot \frac{n \cdot a^{\log_2 n}}{2^{\log_2 n}} = \frac{2}{a} \cdot \frac{n \cdot a^{\log_2 n}}{n} = \frac{2}{a} \cdot a^{\log_2 n}$$

Notice that

$$a^{\log_2 n} = (2^{\log_2 a})^{\log_2 n} = (2^{\log_2 n})^{\log_2 a} = n^{\log_2 a}$$



Total work

- n times the largest term in the geometric series is

$$n \left(\frac{a}{2}\right)^{\log_2 n - 1} = \frac{2}{a} \cdot \frac{n \cdot a^{\log_2 n}}{2^{\log_2 n}} = \frac{2}{a} \cdot \frac{n \cdot a^{\log_2 n}}{n} = \frac{2}{a} \cdot a^{\log_2 n}$$

Notice that

$$a^{\log_2 n} = \left(2^{\log_2 a}\right)^{\log_2 n} = \left(2^{\log_2 n}\right)^{\log_2 a} = n^{\log_2 a}$$

So the total work is

$$a^{\log_2 n} T(1) + n \sum_{i=0}^{\log_2 n - 1} \left(\frac{a}{2}\right)^i$$



Total work

- n times the largest term in the geometric series is

$$n \left(\frac{a}{2}\right)^{\log_2 n - 1} = \frac{2}{a} \cdot \frac{n \cdot a^{\log_2 n}}{2^{\log_2 n}} = \frac{2}{a} \cdot \frac{n \cdot a^{\log_2 n}}{n} = \frac{2}{a} \cdot a^{\log_2 n}$$

Notice that

$$a^{\log_2 n} = \left(2^{\log_2 a}\right)^{\log_2 n} = \left(2^{\log_2 n}\right)^{\log_2 a} = n^{\log_2 a}$$

So the total work is

$$a^{\log_2 n} T(1) + n \sum_{i=0}^{\log_2 n - 1} \left(\frac{a}{2}\right)^i$$

$$\Theta \left(n^{\log_2 a} \right)$$

$$\Theta \left(n^{\log_2 a} \right)$$



Example 5 Recap

■

$$(*) \quad T(n) = \begin{cases} 1 & \text{if } n = 1 \\ 4T(n/2) + n & \text{if } n \geq 2 \end{cases}$$



Example 5 Recap

■

$$(*) \quad T(n) = \begin{cases} 1 & \text{if } n = 1 \\ 4T(n/2) + n & \text{if } n \geq 2 \end{cases}$$

$a = 4$, so the Theorem says that

$$T(n) = \Theta(n^{\log_2 a}) = \Theta(n^{\log_2 4}) = \Theta(n^2)$$



Example 5 Recap

■

$$(*) \quad T(n) = \begin{cases} 1 & \text{if } n = 1 \\ 4T(n/2) + n & \text{if } n \geq 2 \end{cases}$$

$a = 4$, so the Theorem says that

$$T(n) = \Theta(n^{\log_2 a}) = \Theta(n^{\log_2 4}) = \Theta(n^2)$$

This matches with the exact answer of $2n^2 - n$.



Three Different Behaviors

- **Theorem** Suppose that we have a recurrence of the form

$$T(n) = aT(n/2) + n,$$

where a is a positive integer and $T(1)$ is nonnegative. Then we have the following big Θ bounds on the solution:

1. If $a < 2$, then $T(n) = \Theta(n)$.
2. If $a = 2$, then $T(n) = \Theta(n \log n)$.
3. If $a > 2$, then $T(n) = \Theta(n^{\log_2 a})$



The Master Theorem

- **Theorem** Suppose that we have a recurrence of the form

$$T(n) = aT(n/b) + cn^d,$$

where a is a positive integer, $b \geq 1$, c, d are real numbers with c positive and d nonnegative, and $T(1)$ is nonnegative. Then we have the following **big Θ** bounds on the solution:

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



Next Lecture

- counting ...

