南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

DISCRETE MATHEMATICS FOR COMPUTER SCIENCE

Dr. QI WANG

Department of Computer Science and Engineering
Office: Room903, Nanshan iPark A7 Building
Email: wangqi@sustc.edu.cn

Assignment #3

Please submit your assignments before class!

Simple Substitution Ciphers: Examples

Cryptanalysis: Suppose that we know two pairs of plaintext-ciphertext characters (m_1, c_1) and (m_2, c_2), i.e.,

$$c_1 = (m_1 k_0 + k_1) \bmod 26$$
$$c_2 = (m_2 k_0 + k_1) \bmod 26$$

It is possible to solve the equations to obtain k_0 and k_1.

$\mathcal{Q}$: When the set of two equations above has a unique solution (k_0, k_1)?

Simple Substitution Ciphers: Examples

Cryptanalysis: Suppose that we know two pairs of plaintext-ciphertext characters (m_1, c_1) and (m_2, c_2), i.e.,

$$\begin{aligned} c_1 &= (m_1 k_0 + k_1) \bmod 26 \\ c_2 &= (m_2 k_0 + k_1) \bmod 26 \end{aligned}$$

It is possible to solve the equations to obtain k_0 and k_1.

$\mathcal{Q}$: When the set of two equations above has a unique solution (k_0, k_1)?

Case I: $\gcd(m_1 - m_2, 26) = 1$

$$\begin{aligned} k_0 &= (c_1 - c_2)(m_1 - m_2)^{-1} \bmod 26 \\ k_1 &= c_1 - m_1 k_0 \bmod 26 \end{aligned}$$

Case II: $\gcd(m_1 - m_2, 26) = 2$. Suppose that (k_0, k_1), (k_0', k_1') are two solutions. Then $(k_0 - k_0')(m_1 - m_2) \equiv 0 \bmod 26$. Then $k_0' = k_0 + 13\ell$ for $\ell = 0, 1$. Since k_0, k_0' must be coprime to 26, ℓ must be 0 and $k_0 = k_0'$.

Case III: $\gcd(m_1 - m_2, 26) = 13$. $(k_0, k_1) \sim ((k_0 + 2) \bmod 26, k_1)$

RSA Public-Key Cryptosystem

Pick two large primes, p and q. Let $n = pq$, then $\phi(n) = (p-1)(q-1)$. Encryption and decryption keys e and d are selected such that

- $\gcd(e, \phi(n)) = 1$
- $ed \equiv 1 \pmod{\phi(n)}$

- $C = M^e \bmod n$ (RSA **encryption**)
- $M = C^d \bmod n$ (RSA **decryption**)

RSA Proof

- To prove $C^d \bmod n = M$

RSA Proof

- To prove $C^d \bmod n = M$

Case I: $\gcd(M, n) = 1$

By Euler's theorem, $M^{\phi(n)} \equiv 1 \bmod n$. Then

$$\begin{aligned} C^d \bmod n &= M^{ed} \bmod n \\ &= M^{k\phi(n)+1} \bmod n \\ &= (M^{k\phi(n)} \bmod n) M \bmod n \\ &= (M^{\phi(n)} \bmod n)^k M \bmod n \\ &= M, \end{aligned}$$

where k is some integer.

RSA Proof

- To prove $C^d \bmod n = M$

<u>Case II:</u> $\gcd(M, n) = p$

We have $M = tp$, with some integer $0 < t < q$. So $\gcd(M, q) = 1$. Since $ed = k\phi(n) + 1$ for some integer k, by Fermat's theorem, we have

$$(M^{k\phi(n)} - 1) \bmod q = \left((M^{k(p-1)})^{q-1} - 1\right) \bmod q = 0.$$

Whence

$$\begin{aligned}
(M^{ed} - M) \bmod n &= M(M^{ed-1} - 1) \bmod n \\
&= tp\left(M^{k\phi(n)} - 1\right) \bmod pq \\
&= 0
\end{aligned}$$

RSA Proof

- To prove $C^d \bmod n = M$

<u>Case `III`:</u> $\gcd(M, n) = q$

Similar to Case `II`.

<u>Case `IV`:</u> $\gcd(M, n) = pq$

Trivial since $M = C = 0$.

RSA

$\mathcal{Q}$: Consider the RSA system. Let (e, d) be a key pair for the RSA. Define

$$\lambda(n) = \mathrm{lcm}(p-1, q-1)$$

and compute $d' = e^{-1} \bmod \lambda(n)$. Will decryption using d' instead of d still work? (prove $C^{d'} \bmod n = M$)

RSA

$\mathcal{Q}$: Consider the RSA system. Let (e, d) be a key pair for the RSA. Define

$$\lambda(n) = \mathrm{lcm}(p-1, q-1)$$

and compute $d' = e^{-1} \bmod \lambda(n)$. Will decryption using d' instead of d still work? (prove $C^{d'} \bmod n = M$)

<u>Case I:</u> $\gcd(M, n) = 1$

$$\begin{aligned}
C^{d'} \bmod n &= M^{ed'} \bmod n = M^{k\lambda(n)+1} \bmod n \\
&= (M^{k\lambda(n)} \bmod n) M \bmod n \\
&= \left(M^{(p-1)(q-1)/\gcd(p-1,q-1)} \bmod n\right)^k M \bmod n
\end{aligned}$$

By Fermat's theorem, $M^{(p-1)(q-1)/\gcd(p-1,q-1)} \bmod p = \left(M^{(q-1)/\gcd(p-1,q-1)}\right)^{p-1} \bmod p = 1$ and $M^{(p-1)(q-1)/\gcd(p-1,q-1)} \bmod q = 1$. Then by Chinese Remainder Theorem, we have $C^{d'} \bmod n = M$.

$\mathcal{Q}$: Consider the RSA system. Let (e, d) be a key pair for the RSA. Define

$$\lambda(n) = \text{lcm}(p-1, q-1)$$

and compute $d' = e^{-1} \bmod \lambda(n)$. Will decryption using d' instead of d still work? (prove $C^{d'} \bmod n = M$)

<u>Case II:</u> $\gcd(M, n) = p$

$M = tp$ for some integer $0 < t < q$. We have $\gcd(M, q) = 1$ and $ed' = k\lambda(n) + 1$ for some integer k. By Fermat's theorem, we have

$(M^{k\lambda(n)} - 1) \bmod q = (M^{k(p-1)(q-1)/\gcd(p-1,q-1)} - 1) \bmod q = 0.$

Then

$$\begin{aligned}(M^{ed'} - M) \bmod n &= M(M^{ed'-1} - 1) \bmod n \\ &= tp(M^{k\lambda(n)} - 1) \bmod pq \\ &= 0\end{aligned}$$

RSA

$\mathcal{Q}$: Consider the RSA system. Let (e, d) be a key pair for the RSA. Define

$$\lambda(n) = \mathrm{lcm}(p-1, q-1)$$

and compute $d' = e^{-1} \bmod \lambda(n)$. Will decryption using d' instead of d still work? (prove $C^{d'} \bmod n = M$)

<u>Case `III`:</u> $\gcd(M, n) = q$

Similar to Case `II`.

<u>Case `IV`:</u> $\gcd(M, n) = pq$

Trivial.

Growth Rates of Solutions to Recurrences

- Divide and conquer algorithms

- Iteration 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

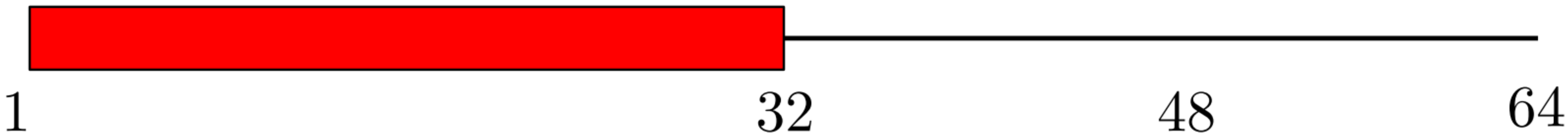
$1 \qquad\qquad\qquad\qquad\qquad 32 \qquad\qquad\qquad 48 \qquad\qquad\qquad 64$

Binary Search Example

1 32 48 64

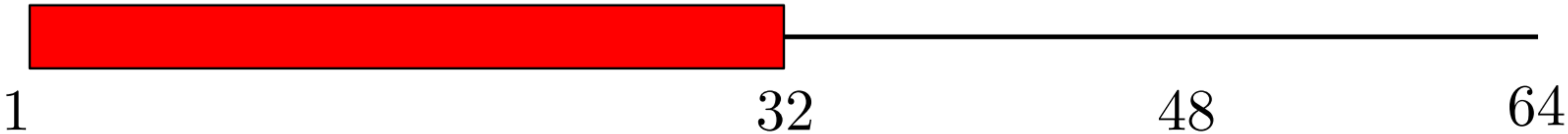
Is $x > 32$?

Binary Search Example



Is $x > 32$? Answer: Yes

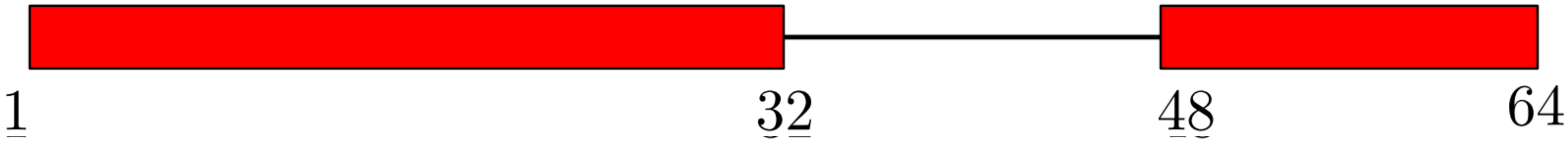
Binary Search Example



Is $x > 32$? Answer: Yes

Is $x > 48$?

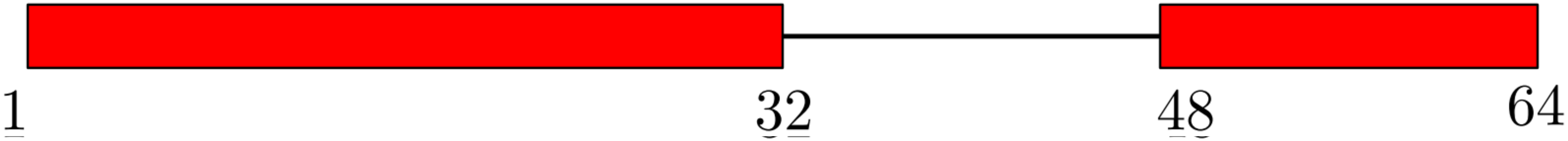
Binary Search Example



Is $x > 32$? Answer: Yes

Is $x > 48$? Answer: No

Binary Search Example

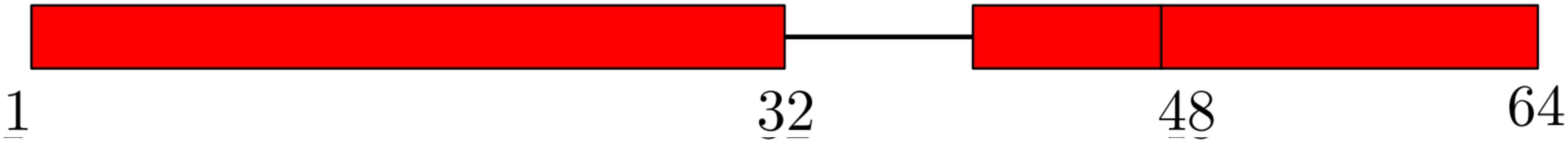


Is $x > 32$? Answer: Yes

Is $x > 48$? Answer: No

Is $x > 40$?

Binary Search Example

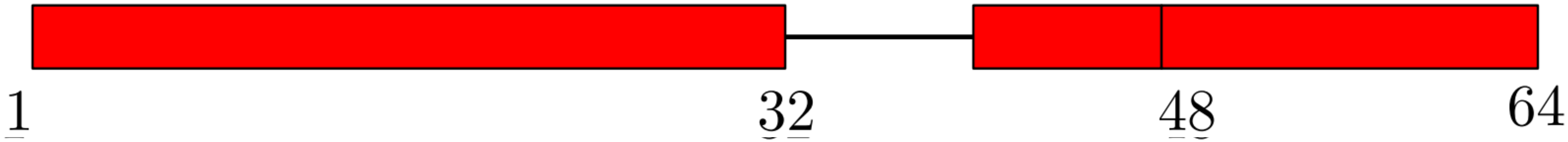


Is $x > 32$? Answer: Yes

Is $x > 48$? Answer: No

Is $x > 40$? Answer: No

Binary Search Example



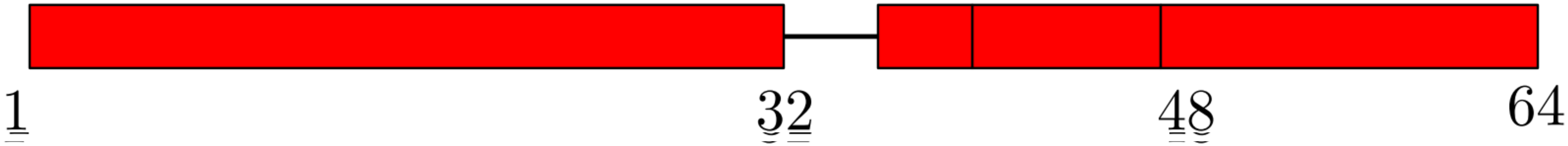
Is $x > 32?$ Answer: Yes

Is $x > 48?$ Answer: No

Is $x > 40?$ Answer: No

Is $x > 36?$

Binary Search Example



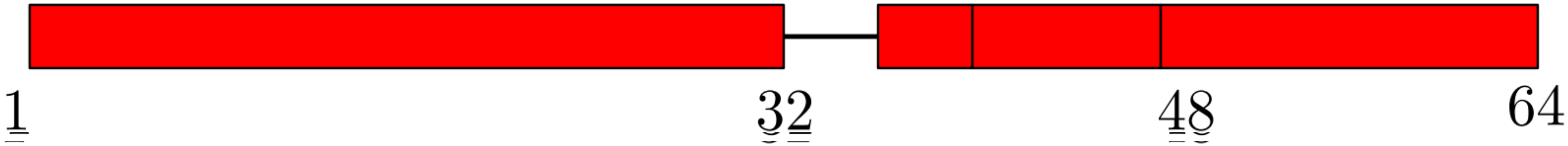
Is $x > 32?$ Answer: Yes

Is $x > 48?$ Answer: No

Is $x > 40?$ Answer: No

Is $x > 36?$ Answer: No

Binary Search Example



Is $x > 32$? Answer: Yes

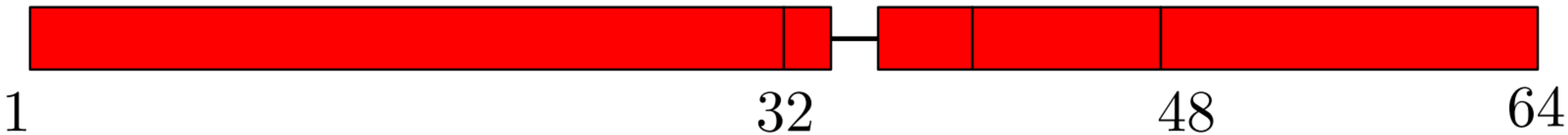
Is $x > 48$? Answer: No

Is $x > 40$? Answer: No

Is $x > 36$? Answer: No

Is $x > 34$?

Binary Search Example



Is $x > 32$? Answer: Yes

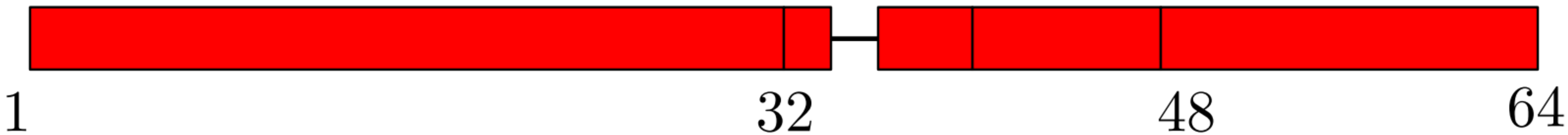
Is $x > 48$? Answer: No

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

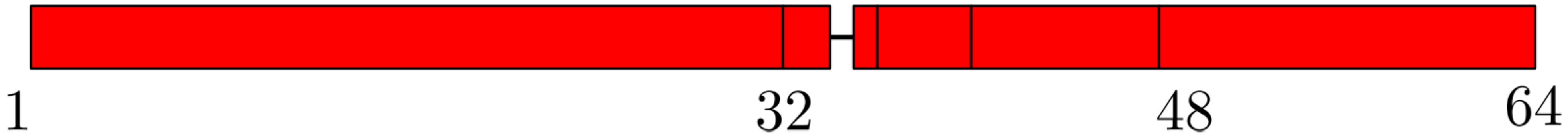
Is $x > 40$? Answer: No

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

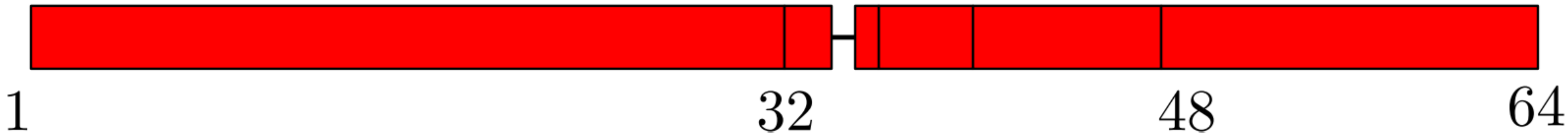
Is $x > 40$? Answer: No

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

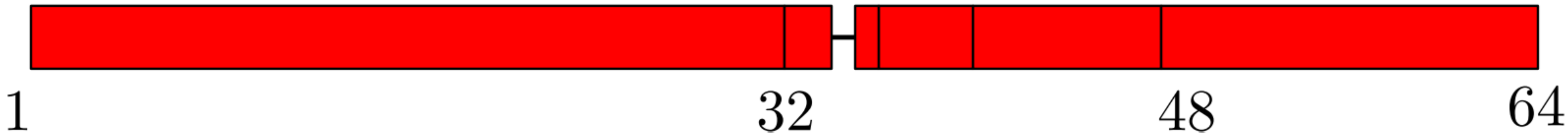
Is $x > 36$? Answer: No

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

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

  For simplicty, 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

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

  For simplicty, 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 (**).

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

Binary Search Example

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

  For simplicty, 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 (**).

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

  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
- Iteration recurrences
- Three different behaviors

Iterating Recurrences

- We will solve recurrence relations by iterating (repeating) them.

Iterating Recurrences

- We will solve recurrence relations by iterating (repeating) them.

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

Iterating Recurrences

- We will solve recurrence relations by iterating (repeating) them.

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

To solve the problem of size n, we
(i) solve 2 subproblems of size $n/2$, and
(ii) do n unites of additional work

Iterating Recurrences

- We will solve recurrence relations by iterating (repeating) them.

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

To solve the problem of size n, we
(i) solve 2 subproblems of size $n/2$, and
(ii) do n unites of additional work

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

Iterating Recurrences

- We will solve recurrence relations by iterating (repeating) them.

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

To solve the problem of size n, we
(i) solve 2 subproblems of size $n/2$, and
(ii) do n unites of additional work

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

To solve the problem of size n, we
(i) solve 1 subproblems of size $n/4$, and
(ii) do n^2 unites of additional work

Iterating Recurrences

- We will solve recurrence relations by iterating (repeating) them.

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

To solve the problem of size n, we
(i) solve 2 subproblems of size $n/2$, and
(ii) do n unites of additional work

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

To solve the problem of size n, we
(i) solve 1 subproblems of size $n/4$, and
(ii) do n^2 unites of additional work

$$T(n) = 3T(n-1) + n$$

Iterating Recurrences

- We will solve recurrence relations by iterating (repeating) them.

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

To solve the problem of size n, we
(i) solve 2 subproblems of size $n/2$, and
(ii) do n unites of additional work

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

To solve the problem of size n, we
(i) solve 1 subproblems of size $n/4$, and
(ii) do n^2 unites of additional work

$$T(n) = 3T(n-1) + n$$

To solve the problem of size n, we
(i) solve 3 subproblems of size $n-1$, and
(ii) do n unites of additional work

Iterating Recurrences: Example 1

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

Iterating Recurrences: Example 1

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

  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

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

  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

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

  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

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

Iterating Recurrences: Example 1

- Algebraically iterating the recurrence

  Assume that n is a power of 2

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

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

Iterating Recurrences: Example 1

- Algebraically iterating the recurrence

  Assume that n is a power of 2

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

Iterating Recurrences: Example 1

- Algebraically iterating the recurrence

  Assume that n is a power of 2

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

Iterating Recurrences: Example 1

- Algebraically iterating the recurrence

  Assume that n is a power of 2

$$
\begin{aligned}
T(n) = 2T\left(\tfrac{n}{2}\right) + n \quad &= 2\left(2T\left(\tfrac{n}{4}\right) + \tfrac{n}{2}\right) + n \\
= 4T\left(\tfrac{n}{4}\right) + 2n \quad &= 4\left(2T\left(\tfrac{n}{8}\right) + \tfrac{n}{4}\right) + 2n \\
= 8T\left(\tfrac{n}{8}\right) + 3n & \\
\vdots \qquad \vdots & \\
= 2^i T\left(\tfrac{n}{2^i}\right) + in & \\
\vdots \qquad \vdots & \\
= 2^{\log_2 n} T\left(\tfrac{n}{2^{\log_2 n}}\right) + (\log_2 n)n &
\end{aligned}
$$

End when $i = \log_2 n$

Iterating Recurrences: Example 1

- Algebraically iterating the recurrence

  Assume that n is a power of 2

$$
\begin{aligned}
T(n) &= 2T\left(\tfrac{n}{2}\right) + n && = 2\left(2T\left(\tfrac{n}{4}\right) + \tfrac{n}{2}\right) + n \\
&= 4T\left(\tfrac{n}{4}\right) + 2n && = 4\left(2T\left(\tfrac{n}{8}\right) + \tfrac{n}{4}\right) + 2n \\
&= 8T\left(\tfrac{n}{8}\right) + 3n \\
&\ \ \vdots \qquad \vdots \\
&= 2^i T\left(\tfrac{n}{2^i}\right) + in \\
&\ \ \vdots \qquad \vdots \\
&= 2^{\log_2 n} T\left(\tfrac{n}{2^{\log_2 n}}\right) + (\log_2 n)n \\
&= nT(1) + n\log_2 n
\end{aligned}
$$

End when $i = \log_2 n$

Iterating Recurrences: Example 1

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

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

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

Iterating Recurrences: Example 1

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

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

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

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

Iterating Recurrences: Example 2

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

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

Iterating Recurrences: Example 2

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

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

Iterating Recurrences: Example 2

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

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

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

Iterating Recurrences: Example 2

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

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

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

Iterating Recurrences: Example 2

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

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

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

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

Iterating Recurrences: Example 2

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

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

Iterating Recurrences: Example 2

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

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

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

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

$$\vdots \qquad \vdots$$

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

$$\vdots \qquad \vdots$$

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

Iterating Recurrences: Example 2

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

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

Iterating Recurrences: Example 3

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

Iterating Recurrences: Example 3

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

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

Iterating Recurrences: Example 3

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

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

Iterating Recurrences: Example 3

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

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

Iterating Recurrences: Example 3

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

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

Iterating Recurrences: Example 3

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

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

Iterating Recurrences: Example 3

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

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

Iterating Recurrences: Example 4

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

Iterating Recurrences: Example 4

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

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

Iterating Recurrences: Example 4

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

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

Iterating Recurrences: Example 4

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

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

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

Iterating Recurrences: Example 4

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

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

Iterating Recurrences: Example 4

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

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

Iterating Recurrences: Example 4

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

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

Iterating Recurrences: Example 4

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

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

Iterating Recurrences: Example 4

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

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

Iterating Recurrences: Example 5

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

Iterating Recurrences: Example 5

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

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

Iterating Recurrences: Example 5

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

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

Iterating Recurrences: Example 5

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

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

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

Iterating Recurrences: Example 5

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

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

$$= 4^2 T\left(\frac{n}{2^2}\right) + \frac{4}{2}n + n \qquad = 4^2\left(4T\left(\frac{n}{2^3}\right) + \frac{n}{2^2}\right) + \frac{4}{2}n + n$$

Iterating Recurrences: Example 5

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

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

$$= 4^2 T\left(\frac{n}{2^2}\right) + \frac{4}{2}n + n \qquad = 4^2\left(4T\left(\frac{n}{2^3}\right) + \frac{n}{2^2}\right) + \frac{4}{2}n + n$$

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

Iterating Recurrences: Example 5

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

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

Iterating Recurrences: Example 5

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

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

$$= 4^2T\left(\frac{n}{2^2}\right) + \frac{4}{2}n + n \qquad = 4^2\left(4T\left(\frac{n}{2^3}\right) + \frac{n}{2^2}\right) + \frac{4}{2}n + n$$

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

$$\vdots \qquad \vdots$$

$$= 4^iT\left(\frac{n}{2^i}\right) + \frac{4^{i-1}}{2^{i-1}}n + \cdots + \frac{4^2}{2^2}n + n$$

$$\vdots \qquad \vdots$$

$$= 4^{\log_2 n}T\left(\frac{n}{2^{\log_2 n}}\right) + \frac{4^{\log_2 n-1}}{2^{\log_2 n-1}}n + \cdots + \frac{4}{2}n + n$$

Iterating Recurrences: Example 5

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

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

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

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

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

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

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

Example 5 Recap

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

Example 5 Recap

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

  $a = 4$, so the Theorem says that

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

Example 5 Recap

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

$a = 4$, so the Theorem says that

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

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 $T(n) = \Theta(n^{\log_b a})$

Some Exercises

- P 13, Ex. 14. Let p, q, and r be the propositions

  p: You get an A on the final exam

  q: You do every exercise in this book

  r: You get an A in this case.

  Write these propositions using p, q and r and logical connectives (including negations).

  c) To get an A in this class, it is necessary for you to get an A on the final.

  d) You get an A on the final, but you don't do every exercise in this book; nevertheless, you get an A in this class.

Some Exercises

- P 13, Ex. 14. Let p, q, and r be the propositions
  p: You get an A on the final exam
  q: You do every exercise in this book
  r: You get an A in this case.
  Write these propositions using p, q and r and logical connectives (including negations).
  c) To get an A in this class, it is necessary for you to get an A on the final.
  d) You get an A on the final, but you don't do every exercise in this book; nevertheless, you get an A in this class.

  Solution.
  c) $r \to p$
  d) $p \wedge \neg q \wedge r$

Some Exercises

- P 16, Ex. 40. Explain, without using a truth table, why $(p \vee \neg q) \wedge (q \vee \neg r) \wedge (r \vee \neg p)$ when p, q and r have the same truth value and it is false otherwise.

Some Exercises

- P 16, Ex. 40. Explain, without using a truth table, why $(p \vee \neg q) \wedge (q \vee \neg r) \wedge (r \vee \neg p)$ when p, q and r have the same truth value and it is false otherwise.

  Proof. Explanation.

Some Exercises

- P 16, Ex. 40. Explain, without using a truth table, why $(p \vee \neg q) \wedge (q \vee \neg r) \wedge (r \vee \neg p)$ when p, q and r have the same truth value and it is false otherwise.

  Proof. Explanation.

  Alternatively, using logical equivalences.

Some Exercises

- P 16, Ex. 40. Explain, without using a truth table, why $(p \vee \neg q) \wedge (q \vee \neg r) \wedge (r \vee \neg p)$ when p, q and r have the same truth value and it is false otherwise.

  Proof. Explanation.

  Alternatively, using logical equivalences.

$$
\begin{aligned}
&(p \vee \neg q) \wedge (q \vee \neg r) \wedge (r \vee \neg p) \\
&\equiv \quad (q \rightarrow p) \wedge (r \rightarrow q) \wedge (p \rightarrow r) \\
&\equiv \quad [(q \rightarrow p) \wedge (r \rightarrow q)] \wedge (p \rightarrow r) \\
&\qquad \wedge (q \rightarrow p) \wedge [(r \rightarrow q) \wedge (p \rightarrow r)] \\
&\equiv \quad (r \leftrightarrow p) \wedge (q \leftrightarrow p) \\
&\equiv \quad r \leftrightarrow p \leftrightarrow q
\end{aligned}
$$

Some Exercises

- P 65, Ex. 10. Let $F(x, y)$ be the statement "x can fool y", where the domain consists of all people in the world. Use quantifiers to express each of these statements.

  g) Nancy can fool exactly two people.

  h) There is exactly one person whom everybody can fool.

  i) No one can fool himself or herself.

  j) There is someone who can fool exactly one person besides himself or herself.

Some Exercises

- P 65, Ex. 10. Let $F(x, y)$ be the statement "x can fool y", where the domain consists of all people in the world. Use quantifiers to express each of these statements.
  g) Nancy can fool exactly two people.
  h) There is exactly one person whom everybody can fool.
  i) No one can fool himself or herself.
  j) There is someone who can fool exactly one person besides himself or herself.

  Solution.
  g) $\exists y_1 \exists y_2 (F(Nancy, y_1) \wedge F(Nancy, y_2) \wedge y_1 \neq y_2 \wedge \forall y (F(Nancy, y) \to (y = y_1 \vee y = y_2)))$
  h) $\exists y (\forall x F(x, y) \wedge \forall z (\forall x F(x, z) \to z = y))$
  i) $\neg \exists x F(x, x)$
  j) $\exists x \exists y (F(x, x) \wedge x \neq y \wedge F(x, y) \wedge \forall z ((F(x, z) \wedge z \neq x) \to z = y))$

Some Exercises

- P 108, Ex. 7. Prove the **triangle inequality**, which states that if x and y are real numbers, then $|x| + |y| \geq |x + y|$.

- P 108, Ex. 7. Prove the **triangle inequality**, which states that if x and y are real numbers, then $|x| + |y| \geq |x + y|$.

Proof.

- Case 1: $x \geq 0$ and $y \geq 0$.
- Case 2: $x < 0$ and $y < 0$.
- Case 3: $x \geq 0$ and $y < 0$.
- Case 4: $x < 0$ and $y \geq 0$.

Some Exercises

- P 108, Ex. 14. Prove or disprove that if a and b are rational numbers, then a^b is also rational.

Some Exercises

- P 108, Ex. 14. Prove or disprove that if a and b are rational numbers, then a^b is also rational.

Proof. Let $a = 2$ and $b = 1/2$.

Some Exercises

- P 109, Ex. 36. Prove that between every two rational numbers there is an irrational number.

Some Exercises

- P 109, Ex. 36. Prove that between every two rational numbers there is an irrational number.

Proof. Let $a = (x + y)/2$.

Some Exercises

- P 126, Ex. 45. Show that if we define the ordered pair (a, b) to be $\{\{a\}, \{a, b\}\}$, then $(a, b) = (c, d)$ if and only if $a = c$ and $b = d$.

Some Exercises

- P 126, Ex. 45. Show that if we define the ordered pair (a, b) to be $\{\{a\}, \{a, b\}\}$, then $(a, b) = (c, d)$ if and only if $a = c$ and $b = d$.

  Proof. We need to show that $\{\{a\}, \{a, b\}\} = \{\{c\}, \{c, d\}\}$ if and only if $a = c$ and $b = d$. The "if" part is straightforward.

  Assume the two sets are equal. We need discuss on two cases: $a \neq b$ and $a = b$.

Some Exercises

- P 137, Ex. 40. Determine whether the symmetric difference is associative, that is, if A, B, C are sets, does it follow that $A \oplus (B \oplus C) = (A \oplus B) \oplus C$?

- P 137, Ex. 40. Determine whether the symmetric difference is associative, that is, if A, B, C are sets, does it follow that $A \oplus (B \oplus C) = (A \oplus B) \oplus C$?

  Proof. Using membership, one can show that each side consists of the elements that are in an odd number of the sets A, B and C.

Some Exercises

- P 137, Ex. 41. Suppose that A, B adn C are sets such that $A \oplus C = B \oplus C$. Must it be the case that $A = B$?

Some Exercises

- P 137, Ex. 41. Suppose that A, B adn C are sets such that $A \oplus C = B \oplus C$. Must it be the case that $A = B$?

Proof. Yes. We prove that for every element $x \in A$, we have $x \in B$ and vice versa.
First, for any element $x \in A$ and $x \notin C$, since $A \oplus C = B \oplus C$, we know that $x \in A \oplus C$ and thus $x \in B \oplus C$. Since $x \notin C$, we must have $x \in B$. For elements $x \in A$ and $x \in C$, we have $x \notin A \oplus C$. Thus, $x \notin B \oplus C$. Since $x \in C$, we must have $x \in B$.

Some Exercises

- P 155, Ex. 70. Show that f is an invertible function from Y to Z and g is an invertible function from X to Y. Show that the inverse of the composition $f \circ g$ is given by $(f \circ g)^{-1} = g^{-1} \circ f^{-1}$.

Some Exercises

- P 155, Ex. 70. Show that f is an invertible function from Y to Z and g is an invertible function from X to Y. Show that the inverse of the composition $f \circ g$ is given by $(f \circ g)^{-1} = g^{-1} \circ f^{-1}$.

  Proof. We have to show that $((f \circ g) \circ (g^{-1} \circ f^{-1}))(z) = z$ for all $z \in Z$ and that $((g^{-1} \circ f^{-1}) \circ (f \circ g))(x) = x$ for all $x \in X$.

Some Exercises

- P 155, Ex. 80. Show that a set S is infinite if and only if there is a proper subset A of S such that there is a one-to-one correspondence between A and S.

Some Exercises

- P 155, Ex. 80. Show that a set S is infinite if and only if there is a proper subset A of S such that there is a one-to-one correspondence between A and S.

  Proof. The "if" part. We prove it by contrapositive. If S is a finite set with cardinality m, then the proper subset A has cardinality strictly smaller than m. So there is no possible bijection between A and S.

  The "only if" part. We try to construct a one-to-one and onto function f from S to A. Let a_0 be one element of S, and let $A = S - \{a_0\}$. Clearly A is infinite. We choose an arbitrary element $a_1 \in A$, and set $f(a_0) = a_1$. For a_1, we choose an arbitrary element $a_2 \in A - \{a_1\}$, and define $f(a_1) = a_2$. Next for a_2, we choose an arbitrary element $a_3 \in A - \{a_1, a_2\}$, and set $f(a_2) = a_3$. Continue this process, and finally let $f(a_i) = a_{i+1}$ for all natural numbers i and $f(x) = x$ for all $x \in S - \{a_0, a_1, a_2, \ldots\}$.

Some Exercises

- P 169, Ex. 38. Derive the formula for $\sum_{k=1}^{n} k^2$.

- P 169, Ex. 38. Derive the formula for $\sum_{k=1}^{n} k^2$.

  Proof. Note that $k^3 - (k-1)^3 = 3k^2 - 3k + 1$. Sum the equation for all values of k from 1 to n.

Some Exercises

- P 169, Ex. 41. Find a formula for $\sum_{k=0}^{m} \lfloor \sqrt{k} \rfloor$, when m is a positive integer.

- P 169, Ex. 41. Find a formula for $\sum_{k=0}^{m}\lfloor\sqrt{k}\rfloor$, when m is a positive integer.

Proof. By the definition of the floor function, there are $2n+1$ n's in the summation. Let $n=\lfloor\sqrt{m}\rfloor-1$. Then

$$
\begin{aligned}
& \sum_{k=0}^{m}\lfloor\sqrt{k}\rfloor \\
= \quad & \sum_{i=1}^{n}(2i^2+i)+(n+1)(m-(n+1)^2+1) \\
= \quad & 2\sum_{i=1}^{n}i^2+\sum_{i=1}^{n}i+(n+1)(m-(n+1)^2+1) \\
= \quad & \frac{n(n+1)(2n+1)}{3}+\frac{n(n+1)}{2}+(n+1)(m-(n+1)^2+1)
\end{aligned}
$$

Some Exercises

- P 176, Ex. 17. If A is an uncountable set and B is a countable set, must $A - B$ be uncountable?

Some Exercises

- P 176, Ex. 17. If A is an uncountable set and B is a countable set, must $A - B$ be uncountable?

  Proof. Consider $A = (A - B) \cup (A \cap B)$.

Some Exercises

- P 217, Ex. 50. Show that if $f(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$, where $a_0, a_1, \ldots, a_{n-1}$, and a_n are real numbers and $a_n \neq 0$, then $f(x)$ is $\Theta(x^n)$.

Some Exercises

- We need to show inequalities in both ways. First, we show that $|f(x)| \le Cx^n$ for all $x \ge 1$ in the following. Noting that $x^i \le x^n$ for such values of x whenever $i < n$. We have the following inequalities, where M is the largest of the absolute values of the coefficients and $C = (n+1)M$: then

$$\begin{aligned} |f(x)| &= |a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0| \\ &\le |a_n|x^n + |a_{n-1}|x^{n-1} + \cdots + |a_1|x + |a_0| \\ &\le |a_n|x^n + |a_{n-1}|x^n + \cdots + |a_1|x^n + |a_0|x^n \\ &\le Mx^n + Mx^n + \cdots + Mx^n \\ &= Cx^n. \end{aligned}$$

For the other direction, let k be chosen larger than 1 and larger than $2nm/|a_n|$, where m is the largest of the absolute values of the a_i's for $i < n$. Then each a_{n-i}/x^i will be smaller than $|a_n|/2n$ in absolute value for all $x > k$. Now we have for all $x > k$,

$$\begin{aligned} |f(x)| &= |a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0| \\ &= x^n \left| a_n + \frac{a_{n-1}}{x} + \cdots + \frac{a_1}{x^{n-1}} + \frac{a_0}{x^n} \right| \\ &\ge x^n |a_n/2|. \end{aligned}$$

Some Exercises

- P 218, Ex. 71. Show that $n \log n$ is $O(\log n!)$.

Some Exercises

- P 218, Ex. 71. Show that $n \log n$ is $O(\log n!)$.

Proof. Try to prove that $(n!)^2 \geq n^n$.

Some Exercises

- P 218, Ex. 71. Show that $n \log n$ is $O(\log n!)$.

Proof. Try to prove that $(n!)^2 \geq n^n$.

Use $(n-i)(i+1) \geq n$ for $i = 0, 1, \ldots, n-1$.

Some Exercises

- P 272, Ex. 11. Show that $\log_2 3$ is an irrational number.

Some Exercises

- P 272, Ex. 11. Show that $\log_2 3$ is an irrational number.

Proof. Suppose that $\log_2 3 = a/b$ where $a, b \in \mathbf{Z}^+$ and $b \neq 0$. Then $2^{a/b} = 3$, so $2^a = 3^b$. This violates the fundamental theorem of arithmetic. Hence $\log_2 3$ is irrational.

Some Exercises

- P 274, Ex. 55. Show that there are infinitely many primes of the form $4k + 3$, where k is a nonnegative integer.

- P 274, Ex. 55. Show that there are infinitely many primes of the form $4k+3$, where k is a nonnegative integer.

  Proof. Suppose that there are only finitely many primes of the form $4k+3$, namely $p_1, p_2, \ldots, p_n$, where $p_1 = 3, p_2 = 7$ and so on. Let $P = 4p_1p_2\cdots p_n - 1$. Note that P is of the form $4k+3$. If P is prime, then we have found a prime of the desired form different from all those listed. If P is not prime, then P has at least one prime factor not in the list $p_1, p_2, \ldots, p_n$, because the remainder when P is divided by p_j is nonzero. Since all odd primes are either of the form $4k+1$ or $4k+3$, and the product of primes of the form $4k+1$ is also of the form $4k+1$, there must be a factor of P of the form $4k+3$ different from the primes we listed.

Announcements

- **Homework assignment** 4
  - ◇ P330 Ex. 25, 26, P331 Ex. 44, P341 Ex. 4, P344 Ex. 37, 42, P371 Ex. 24, 26, P535 Ex. 12, 22, P536 Ex. 34, 36
  - ◇ Due on *Nov. 21st, 2017 before class*
  - ◇ Please try you best to slove problems marked with $*$
  - ◇ Please write your homeowrk **neatly**, as a courtesy to graders.

Next Lecture

- counting, ...