In-Class Problems Week 5, Fri.

Problem 1.
Find the remainder of $26^{1818181}$ divided by 297. *Hint:* $1818181 = (180 \cdot 10101) + 1$; Euler's theorem

Problem 2.
Find an integer $k > 1$ such that n and n^k agree in their last three digits whenever n is divisible by neither 2 nor 5. *Hint:* Euler's theorem.

Problem 3.
Suppose a, b are relatively prime and greater than 1. In this problem you will prove the *Chinese Remainder Theorem*, which says that for all m, n, there is an x such that

$$x \equiv m \bmod a, \tag{1}$$
$$x \equiv n \bmod b. \tag{2}$$

Moreover, x is unique up to congruence modulo ab, namely, if x' also satisfies (1) and (2), then

$$x' \equiv x \bmod ab.$$

(a) Prove that for any m, n, there is some x satisfying (1) and (2).
Hint: Let b^{-1} be an inverse of b modulo a and define $e_a ::= b^{-1}b$. Define e_b similarly. Let $x = me_a + ne_b$.

(b) Prove that

$$[x \equiv 0 \bmod a \text{ AND } x \equiv 0 \bmod b] \quad \text{implies} \quad x \equiv 0 \bmod ab.$$

(c) Conclude that

$$[x \equiv x' \bmod a \text{ AND } x \equiv x' \bmod b] \quad \text{implies} \quad x \equiv x' \bmod ab.$$

wed's classi

(d) Conclude that the Chinese Remainder Theorem is true.

(e) What about the converse of the implication in part (c)?

Problem 4.
Suppose a, b are relatively prime integers greater than 1. In this problem you will prove that Euler's function is *multiplicative*, namely, that

$$\phi(ab) = \phi(a)\phi(b).$$

The proof is an easy consequence of the Chinese Remainder Theorem .

(a) Conclude from the Chinese Remainder Theorem that the function $f : [0, ab) \to [0, a) \times [0, b)$ defined by

$$f(x) ::= (\text{rem}(x, a), \text{rem}(x, b))$$

is a bijection.

(b) For any positive integer, k, let k^* be the integers in $[1, k)$ that are relatively prime to k. Prove that the function f from part (a) also defines a bijection from $(ab)^*$ to $a^* \times b^*$.

(c) Conclude from the preceding parts of this problem that

$$\phi(ab) = \phi(a)\phi(b). \tag{3}$$

(d) Prove Corollary ??: for any number $n > 0$, if $p_1, p_2, \ldots, p_j$ are the (distinct) prime factors of n, then

$$\phi(n) = n\left(1 - \frac{1}{p_1}\right)\left(1 - \frac{1}{p_2}\right)\cdots\left(1 - \frac{1}{p_j}\right).$$

In Class Problem

1 our board)

$297 = 3 \cdot 3 \cdot 3 \cdot 11$

$\phi(297) = (3^2 - 3^2) \cdot (11 - 1) = 18 \cdot 10 = 180$

by Euler's Theorm

$26 = 2 \cdot 13 \Rightarrow GCD(26, 297) = 1$
pair must be relativly prime

$26^{180} = 26^{\phi(297)} \equiv 1 \pmod{297}$

$26^{181818} = 26(26^{180})^{1010}$

$26 \cdot (26^{180})^{10101} \equiv 26 \cdot (1^{10101}) \pmod{297}$

$26^{1818181} \equiv 26 \pmod{297}$

So $\text{rem}(26^{1818181}, 297) = 26$

---

2. For mod k, to leave the last 3 digits unchanged $k = \phi(1000)$
Since n is not divisible by 2 or 5, n is relativly prime to 1000, then by Euler's thorm $k^{\phi(n-1)} \equiv 1 \text{ mod } 1000$

$k^{\phi(1000)} = 1 \text{ mod } 1000$

$n^{\phi(1000)+1} \equiv n \text{ mod } 1000$

(2)

2 Board Rewrite

To have the last 3 digits the same we need $n \equiv n^k \mod 1000$ since n is not divisible by 2 or 5, n is relatively prime to 1000, then by Euler's theorm:

$$n^{\phi(1000)} \equiv 1 \mod 1000$$

$$n^{\phi(1000)+1} \equiv n \mod 1000$$

So $k = \phi(1000)+1$

$$\phi(1000) = (2^3 - 2^2)(5^3 - 5^2)$$

$$= 400$$

$k = 401$

(3)

a) [board]

$e_a ::= b^{-1}b \pmod a$

$e_b ::= a^{-1}a \pmod b$ — since relatively prime

$x = m e_a + n e_b$

$\equiv m \underbrace{b^{-1}b}_{1} + 0 \pmod a$ — inverse becomes 1

$\equiv m \pmod a$

$x \equiv 0 + n \underbrace{a^{-1}a}_{1} \pmod b$

$\equiv n \pmod b$

Thus x exists and $\gcd(a,b) = 1$

b) $x \equiv 0 \mod a \rightarrow a | x$

$x \equiv 0 \mod b \rightarrow b | x \Rightarrow ab | x$

$x = 0 \mod ab$

(4)

3d (board)) Let $y = x - x'$

$y = (x - x') \equiv (x' - x') = 0 \pmod a$

$(x - x') \equiv (x' - x') \pmod b$

$\overset{"}{0}$

$x = y + x'$

---

by part b

$y \equiv 0 \pmod a$ and $y \equiv 0 \pmod b \rightarrow y \equiv 0 \pmod{ab}$

$x' < a$ and $x' < b$, So $x' > a, b$ $\quad a, b \in \mathbb{N}$

Therefore, $x' \equiv x' \pmod a$ and $x' \equiv x' \pmod b \Rightarrow x' \equiv x' \bmod ab$

Consequently if

$x = y + x' \equiv 0 \pmod a$ and $x \equiv 0 \pmod b$

$y + x' \equiv 0 + y \pmod{ab}$

So $x \equiv x' \pmod{ab}$

Massachusetts Institute of Technology
6.042J/18.062J, Spring '11: Mathematics for Computer Science — March 4
Prof. Albert R Meyer — revised Wednesday 2nd March, 2011, 18:41

Solutions to In-Class Problems Week 5, Fri.

Problem 1.
Find the remainder of $26^{1818181}$ divided by 297. *Hint:* $1818181 = (180 \cdot 10101) + 1$; Euler's theorem

Solution. 26.
Since $26 = 2 \cdot 13$ and $297 = 3^3 \cdot 11$ are relatively prime, Euler's theorem implies that

$$k^{\phi(297)} \equiv 1 \pmod{297}$$

where

$$\begin{aligned}
\phi(297) &= \phi(3^3 \cdot 11) \\
&= \phi(3^3) \cdot \phi(11) && \text{(since } \gcd(3^3, 11) = 1) \\
&= (3^3 - 3^2) \cdot (11 - 1) && \text{(since 3 and 11 are prime)} \\
&= 180.
\end{aligned}$$

Using the hint that $1818181 = (180 \cdot 10101) + 1$, we can conclude

$$\begin{aligned}
26^{1818181} &= 26 \cdot 26^{180 \cdot 10101} \\
&\equiv 26 \cdot 1^{10101} \pmod{297} && \text{(by Euler's Theorem)} \\
&= 26.
\end{aligned}$$

■

Problem 2.
Find an integer $k > 1$ such that n and n^k agree in their last three digits whenever n is divisible by neither 2 nor 5. *Hint:* Euler's theorem.

Solution. Two numbers agree in their last three digits iff they are congruent modulo 1000. So we must find a $k > 1$ such that

$$n \equiv n^k \pmod{1000}$$

for all n not divisible by 2 or 5—that is, for all n relatively prime to 1000. But by Euler's theorem, we know $k = \phi(1000) + 1$ will work, namely,

$$k = \phi(1000) + 1 = \phi(2^3)\phi(5^3) + 1 = 4 \cdot 100 + 1 = 401.$$

■

documentclass[problem]mcs

Problem 3.
Suppose a, b are relatively prime and greater than 1. In this problem you will prove the *Chinese Remainder Theorem*, which says that for all m, n, there is an x such that such that

$$x \equiv m \text{ mod } a, \qquad (1)$$
$$x \equiv n \text{ mod } b. \qquad (2)$$

Moreover, x is unique up to congruence modulo ab, namely, if x' also satisfies (1) and (2), then

$$x' \equiv x \text{ mod } ab.$$

(a) Prove that for any m, n, there is some x satisfying (1) and (2).
Hint: Let b^{-1} be an inverse of b modulo a and define $e_a ::= b^{-1}b$. Define e_b similarly. Let $x = me_a + ne_b$.

Solution. We have by definition

$$e_a ::= b^{-1}b \equiv \begin{cases} 1 \text{ mod } a, \\ 0 \text{ mod } b, \end{cases}$$

and likewise for e_b. Therefore

$$me_a + ne_b \equiv \begin{cases} m \cdot 1 + n \cdot 0 = m \text{ mod } a \\ m \cdot 0 + n \cdot 1 = n \text{ mod } b. \end{cases}$$

■

(b) Prove that

$$[x \equiv 0 \text{ mod } a \text{ AND } x \equiv 0 \text{ mod } b] \quad \text{implies} \quad x \equiv 0 \text{ mod } ab.$$

Solution. If $x \equiv 0$ mod a, then by definition, $a \mid x$. Likewise, $b \mid x$. But a and b are relatively prime, so by Unique Factorization **??**, $ab \mid x$, that is, $x \equiv 0$ mod ab. ■

(c) Conclude that

$$\left[x \equiv x' \text{ mod } a \text{ AND } x \equiv x' \text{ mod } b\right] \quad \text{implies} \quad x \equiv x' \text{ mod } ab.$$

Solution. $(x' - x)$ is $\equiv 0$ mod a by (1) and $\equiv 0$ mod b by (2), so by part (b), $(x' - x) \equiv 0$ mod ab. Adding x to both sides of this $\equiv$ gives

$$x' \equiv x \text{ mod } ab.$$

■

(d) Conclude that the Chinese Remainder Theorem is true.

Solution. The existence of an x is given in part (a), so all that's let is to prove x is unique up to congruence modulo ab. But if x and x' both satisfy (1) and (2), then $x' \equiv x$ mod a and $x' \equiv x$ mod a, so $x' \equiv x$ mod ab by part (c). ■

(e) What about the converse of the implication in part (c)?

Solution. The converse is true too: if $cd \mid (x' - x)$, then obviously $c \mid (x' - x)$. This means that

$$x' \equiv x \text{ mod } cd \quad \text{implies} \quad x' \equiv x \text{ mod } c.$$

So in particular,

$$x \equiv x' \text{ mod } ab \quad \text{implies} \quad \left[x \equiv x' \text{ mod } a \text{ AND } x \equiv x' \text{ mod } b\right].$$

So this together with part (c) gives a basic fact worth calling a

Lemma. *For a, b are relatively prime and greater than 1,*

$$\left[x' \equiv x \text{ mod } a \text{ AND } x' \equiv x \text{ mod } b\right] \quad \textit{iff} \quad x' \equiv x \text{ mod } ab.$$

■

Problem 4.
Suppose a, b are relatively prime integers greater than 1. In this problem you will prove that Euler's function is *multiplicative*, namely, that

$$\phi(ab) = \phi(a)\phi(b).$$

The proof is an easy consequence of the Chinese Remainder Theorem .

(a) Conclude from the Chinese Remainder Theorem that the function $f : [0, ab) \to [0, a) \times [0, b)$ defined by

$$f(x) ::= (\text{rem}(x, a), \text{rem}(x, b))$$

is a bijection.

Solution. The Chinese Remainder Theorem says that the congruences

$$x \equiv m \quad (\text{mod } a),$$
$$x \equiv n \quad (\text{mod } b).$$

have a solution $x \in [0, ab)$, which means that f is surjective, and that the solution is unique, which means that f is injective, and hence it is a bijection. ■

(b) For any positive integer, k, let k^* be the integers in $[1, k)$ that are relatively prime to k. Prove that the function f from part (a) also defines a bijection from $(ab)^*$ to $a^* \times b^*$

Solution. But since a and b are relatively prime, number x is relatively prime to ab iff x is relatively prime to a and x is relatively prime to b, by Unique Factorization. This means precisely that $x \in (ab)^*$ iff $f(x) \in a^* \times b^*$, which in turn means $f((ab)^*) = a^* \times b^*$. So restricting the bijection, f, to codomain $(ab)^*$ defines a bijection to $a^* \times b^*$. ■

(c) Conclude from the preceding parts of this problem that

$$\phi(ab) = \phi(a)\phi(b). \tag{3}$$

Solution. The mapping f defines a bijection between $(ab)^*$ and $a^* \times b^*$. So

$$\phi(ab) ::= |(ab)^*| = |a^* \times b^*| = |a^*| \cdot |b^*| = \phi(a) \cdot \phi(b).$$

■

(d) Prove Corollary **??**: for any number $n > 0$, if $p_1, p_2, \ldots, p_j$ are the (distinct) prime factors of n, then

$$\phi(n) = n\left(1 - \frac{1}{p_1}\right)\left(1 - \frac{1}{p_2}\right)\cdots\left(1 - \frac{1}{p_j}\right).$$

Solution. We know from Theorem **??** that for all primes, p, and $k > 0$,

$$\phi(p^k) = p^k - p^{k-1} = p^k\left(1 - \frac{1}{p}\right).$$

So if

$$n = p_1^{k_1} p_2^{k_2} \cdots p_j^{k_j}$$

where all the k's are positive, then repeated applications of (3) we get

$$\begin{aligned}
\phi(n) &= \phi(p_1^{k_1}) \cdot \phi(p_2^{k_2}) \cdots \phi(p_j^{k_j}) \\
&= p_1^{k_1}\left(1 - \frac{1}{p_1}\right) \cdot p_1^{k_2}\left(1 - \frac{1}{p_2}\right) \cdots p_1^{k_j}\left(1 - \frac{1}{p_j}\right) \\
&= p_1^{k_1} \cdot p_2^{k_2} \cdots p_j^{k_j} \cdot \left(1 - \frac{1}{p_1}\right)\left(1 - \frac{1}{p_2}\right) \cdots \left(1 - \frac{1}{p_j}\right) \\
&= n\left(1 - \frac{1}{p_1}\right)\left(1 - \frac{1}{p_2}\right) \cdots \left(1 - \frac{1}{p_j}\right).
\end{aligned}$$

■

Tutor

(late, last week's problems)

TP 5.1 GCDs

What is the GCD(21212121, ~~1212~~ 12121212)

So ~~rem~~

GCD(12121212, rem(21212121, 12121212))

~~GCD 2 12~~

GCD(9090909, rem(12121212, 9090909))

GCD(3030303, rem(9090909, 3030303))

GCD(~~30~~ 3030303, 0)

↑ ✓

b) How many steps? ~~4~~ 3

TP2 GCDs 2

Compute GCD

$X = 17^{88} \cdot 31^{5} \cdot 37^{2} \cdot 59^{1000}$

$Y = 19^{(922)} \cdot 37^{12} \cdot 53^{3678} \cdot 59^{29}$

(2)

Give the prime factorization as a set of prime exponent pairs

↓ The # that multiply together to make it
prime
(unique I believe)
Fund. Theorm of algebra

$60 = 2^2 \cdot 3 \cdot 5$

is ~~(3 1)~~

(3 1)(2 2)(5 1)

?. Is there a shortcut way to do this?

I could not find in book

Tried Wolfram alpha - got overflow

Give up (37 2) (59 29)

Iterate over all the primes that exist in both factorizations

Raise each of them to the smallest of 2 exponents

Then multiply the resulting powers

If you replace smallest w/ Greatest get LCM

(Least common multiple)

So 59^{29} ~~$59^{29} \cdot 59^{29}$~~ and 31^5

So $59^{29} \cdot 31^5$ is GCD

- where did we learn that trick - I kinda remember it

③

TP. 3 Divisors

How many prime divisors does 12 have

1 2 3 4 6 12
P P P 2 (1 is not prime)

Positive ?
6 ✓

total ?
12 ✓

TP 5.4 Divisability + Congruence

List the equivilent statements

1. $a \equiv b \pmod{n}$ ← I am guessing this is the base case

4 $n \mid (a-b)$

3. $Rem(a,n) = rem(b,n)$

5. $a = b + nk$ some k

6. $(a-b)$ is multiple of n ✓

Not $a=b$

$n \mid a$ or $n \mid b$

(4)

5.5 Multiplicative Inverse

(mod 7) of 2

- did in p-set

F like little theorm

$k^{p-2} \cdot k \equiv 1 \pmod{p}$

?. multiplicate inverse means 1

$2^0 \cdot 2 = 1 \pmod{2}$

rem $(2^0, 2)$

1 so 1 is ans (x)

I guess I forgot the process

$\_\_ \cdot 2 \equiv 1 \mod 7$

Guess + check

$n = 1$

~~x~~ $1 \cdot 2 \pmod{7} = 2$

$n = 2$ X

$n = 3$ X

$n = 4$ ✓ ~~x~~ $4 \cdot 2 = 8 \mod (7) = 1$

(5) # TP 5.6 Linear Combinations + Inverses

Part 1 GCDs w/ Linear Combos

Find x, y $\quad 25x + 32y = GCD(25, 32)$

So they want us to pulverize!

	$x - q \cdot y$	
$GCD(32, rem(25, 32))$	$25 = 32 - 0 \cdot 25$	
$GCD(25, rem(32, 25))$	$7 = 32 - 1 \cdot 25$	$= 32 - 1(32 - 0 \cdot 25)$ $= 1 \cdot 32 - 1 \cdot 25$
$GCD(7, rem(25, 7))$	$4 = 25 - 3 \cdot 7$	$= (32 - 0 \cdot 25) - 3(1 \cdot 32 - 1 \cdot 25)$ $= 4 \cdot 25 - 3 \cdot 32$
$GCD(4, rem(7, 4))$	$3 = 7 - 1 \cdot 4$	$= (32 - 25) - 1(4 \cdot 25 - 3 \cdot 32)$ $= 4 \cdot 32 - 5 \cdot 25$
$GCD(3, rem(4, 3))$	$1 = 4 - 1 \cdot 3$	$= (4 \cdot 25 - 3 \cdot 32) - 1(4 \cdot 32 - 5 \cdot 25)$ $= 9 \cdot 25 - 7 \cdot 32$
$GCD(1, rem(1, 3))$	$1 = 1 - 0 \cdot 3$	$= (9 \cdot 25 - 7 \cdot 32) - 0 \cdot \_$
$GCD(1, rem(1, 1))$	$0 = 1 - 1 \cdot 1$	
$GCD(1, 0)$ ↑ 1		

So $(9, -7)$ ☺

Thats actually really cool - first time actually did it

⑥

Part 2: Inverse w/ Linear Comb

What is inverse (mod 25) of 32

$32 \cdot \_ \equiv 1 \pmod{25}$

Let me try Fermat's Little Theorm

$k^{p-2} \equiv \_ \pmod{p}$

$32^{23} \equiv \_\_\_ \pmod{25}$

Fast exponentation math done mod 25

$X = 32$

$Y = 1$

$b = 23$

$r = rem(23,2) \bmod 25 = 1$

$z = quot(23,2) \bmod 25 = 11$

$y = xy \bmod 25 = 7$

$X = x^2 \bmod 25 = 32^2 = 24$ ← do then $\overline{25}$ and take remainder

$r = rem(11,2) = 1$

$z = \ldots = 5$

$Y = 24 \cdot 7 \bmod 25 = 18$

$X = x^2 = 24^2 \bmod 25 = 1$

(7)

r = rem (5, 2) = 1

z = quot (5, 2) = 2

y = xy = 18·1 = 18

x = 1^2 = 1

r = rem (2, 2) = 0

z = quot (2, 2) = 0

x = 1

return 18 ✓

TP 5.7 Fermat's Little Theorm

What is Rem(24^{78}, 79)

- how is this his theorm
- oh fast exponentiation mod 79
- just did

Will cheat on this, since I just did it

1 ✓

Since 79 is prime and 24 is not a multiple of 79,
F Little T applicable 24^{79-1} is congruent to 1 mod 79

Oh I did not realize 78 was p-1
I always think of it by p-2 / fast exponetiation

TP 5.8 Eulers Theorm

What is $\phi(175)$

- So # of integers relativly prime to 175
  - where $\gcd(a, 175) = 1$

It also has a more user friendly def

that are not cancelable ie $\frac{175}{5} \neq$ integer

- must be better way to say that

¿So how to compute

- Can do products of two primes
- but don't know, since Factorization is hard

So - Theorm 8.7.6

$\phi(p^k)$ or $\phi(ab)$ for relativly prime a, b

So could do prime factorization

$5 \cdot 5 \cdot 7$

So $\phi(5) \cdot \phi(5) \cdot \phi(7)$

Or $\phi(5^2) \cdot \phi(7)$

$(5^2 - 5^1) \cdot (7^1 - 7^0)$
↑ must be prime

$(25 - 5) \cdot (7 - 1) = 20 \cdot 6 = 120$ ✓

(9)

b) What is rem $(22^{12001}, 175)$

How find this easily?

How related to ~~speci~~ this (Euler) question

The exponent of k need to produce an inverse of k mod n relies on $\phi(n)$

$$k^{\phi(n)} \equiv 1 \pmod n$$

So

$$22^{12001} = 1 \pmod{175}$$

$12201 = \phi(175)$ – no!

~~k has~~ k has to be rel. prime to n

$k^{\phi(n)-1}$ is multiplicative inverse of k mod n

I am not seeing the connection here...

~~[illegible]~~ $22^{12001} = \_\_ \pmod{175}$

How does Euler tie in?

$22^{12001} = 22^{(120 \cdot 100)+1} = (22^{120})^{100} \cdot 22 \equiv 1^{100} \cdot 22$

$\equiv 22 \bmod 175$

Cheat 22 ✓

thought the 1 looked suspicious

(10)

TP 5.4 ~~the~~ Relative Primality

How many # b/w 1, 3780 are relatively prime to 3780

I could find total $\phi(3780)$

Oh is what they want - did not see

First prime factor

- which computer must have precomputed

$2^2 \cdot 3^3 \cdot 5 \cdot 7$

$\phi(2^2) \cdot \phi(3^3) \cdot \phi(5) \cdot \phi(7)$

$(2^2-2^1)(3^3-3^2)(5^1-5^0)(7^1-7^0)$

$(4-2)(27-9)(5-1)(7-1)$

$2 \cdot 18 \cdot 4 \cdot 6$

864

In-Class Problems Week 6, Mon.

Problem 1.
Let's try out RSA! There is a complete description of the algorithm in the text box. You'll probably need extra paper. **Check your work carefully!**

(a) As a team, go through the **beforehand** steps.

- Choose primes p and q to be relatively small, say in the range 10-40. In practice, p and q might contain several hundred digits, but small numbers are easier to handle with pencil and paper.
- Try $e = 3, 5, 7, \dots$ until you find something that works. Use Euclid's algorithm to compute the gcd.
- Find d (using the Pulverizer—see appendix for a reminder on how the Pulverizer works—or Euler's Theorem).

When you're done, put your public key on the board. This lets another team send you a message.

(b) Now send an encrypted message to another team using their public key. Select your message m from the codebook below:

- 2 = Greetings and salutations!
- 3 = Yo, wassup?
- 4 = You guys are slow!
- 5 = All your base are belong to us.
- 6 = Someone on *our* team thinks someone on *your* team is kinda cute.
- 7 = You *are* the weakest link. Goodbye.

(c) Decrypt the message sent to you and verify that you received what the other team sent!

832

p=17
q=53
n= ~~[illegible]~~ 901
e=3
d= ~~633~~ 555
↑ figure out d

The RSA Cryptosystem

Beforehand The receiver creates a public key and a secret key as follows.

1. Generate two distinct primes, p and q. Since they can be used to generate the secret key, they must be kept hidden.
2. Let $n = pq$.
3. Select an integer e such that $\gcd(e, (p-1)(q-1)) = 1$.
   The *public key* is the pair (e, n). This should be distributed widely.
4. Compute d such that $de \equiv 1 \pmod{(p-1)(q-1)}$. This can be done using the Pulverizer.
   The *secret key* is the pair (d, n). This should be kept hidden!

Encoding Given a message m, the sender first checks that $\gcd(m, n) = 1$.

The sender then encrypts message m to produce m^* using the public key:

$$m^* = \text{rem}(m^e, n).$$

Decoding The receiver decrypts message m^* back to message m using the secret key:

$$m = \text{rem}((m^*)^d, n).$$

Problem 2.

A critical fact about RSA is, of course, that decrypting an encrypted message always gives back the original message! That is, that $\text{rem}((m^d)^e, pq) = m$. This will follow from something slightly more general:

Lemma 2.1. *Let n be a product of distinct primes and $a \equiv 1 \pmod{\phi(n)}$ for some nonnegative integer, a. Then*

$$m^a \equiv m \pmod{n}. \tag{1}$$

(a) Explain why Lemma 2.1 implies that k and k^5 have the same last digit. For example:

$$\underline{2}^5 = 3\underline{2} \qquad 7\underline{9}^5 = 307705639\underline{9}$$

Hint: What is $\phi(10)$?

(b) Explain why Lemma 2.1 implies that the original message, m, equals $\text{rem}((m^e)^d, pq)$.

(c) Prove that if p is prime, then

$$m^a \equiv m \pmod{p} \tag{2}$$

for all nonnegative integers $a \equiv 1 \pmod{p-1}$.

(d) Prove that if $a \equiv b \pmod{p_i}$ for distinct primes $p_1, p_2, \ldots, p_n$, then $a \equiv b \pmod{p_1 p_1 \cdots p_n}$.

(e) Combine the previous parts to complete the proof of Lemma 2.1.

Appendix

Inverses, Fermat, Euler

Lemma (Inverses mod n). *If k and n are relatively prime, then there is integer k' called the* modulo n inverse *of k, such that*

$$k \cdot k' \equiv 1 \pmod{n}.$$

Remark: If $\gcd(k,n) = 1$, then $sk + tn = 1$ for some s, t, so we can choose $k' ::= s$ in the previous Lemma. So given k and n, an inverse k' can be found efficiently using the Pulverizer.

Theorem (Fermat's (Little) Theorem). *If p is prime and k is not a multiple of p, then*

$$k^{p-1} \equiv 1 \pmod{p}$$

Definition. The value of *Euler's totient function*, $\phi(n)$, is defined to be the number of positive integers less than n that are relatively prime to n.

Lemma (Euler Totient Function Equations).

$$\phi(p^k) = p^k - p^{k-1} \qquad \text{for prime, } p, \text{ and } k > 0,$$
$$\phi(mn) = \phi(m) \cdot \phi(n) \qquad \text{when } \gcd(m,n) = 1.$$

Theorem (Euler's Theorem). *If k and n are relatively prime, then*

$$k^{\phi(n)} \equiv 1 \pmod{n}$$

The Pulverizer

Euclid's algorithm for finding the GCD of two numbers relies on repeated application of the equation:

$$\gcd(a,b) = \gcd(b, \text{rem}(a,b))$$

The Pulverizer goes through the same steps, but requires some extra bookkeeping along the way: as we compute $\gcd(a,b)$, we keep track of how to write each of the remainders (49, 21, and 7, in the example) as a linear combination of a and b (this is worthwhile, because our objective is to write the last nonzero remainder, which is the GCD, as such a linear combination). For our example, here is this extra bookkeeping:

x	y	$\text{rem}(x,y)$	$=$	$x - q \cdot y$
259	70	49	$=$	$259 - 3 \cdot 70$
70	49	21	$=$	$70 - 1 \cdot 49$
			$=$	$70 - 1 \cdot (259 - 3 \cdot 70)$
			$=$	$-1 \cdot 259 + 4 \cdot 70$
49	21	7	$=$	$49 - 2 \cdot 21$
			$=$	$(259 - 3 \cdot 70) - 2 \cdot (-1 \cdot 259 + 4 \cdot 70)$
			$=$	$\boxed{3 \cdot 259 - 11 \cdot 70}$
21	7	0		

We began by initializing two variables, $x = a$ and $y = b$. In the first two columns above, we carried out Euclid's algorithm. At each step, we computed $\text{rem}(x,y)$, which can be written in the form $x - q \cdot y$. Then we replaced x and y in this equation with equivalent linear combinations of a and b, which we already had computed. After simplifying, we were left with a linear combination of a and b that was equal to the remainder as desired. The final solution is boxed.

6.042 In-Class Np Front

3/7

(5 min late)

Key Generation

$p = 17$

$q = 31$ so $n = 527$

picked arbitrary

$e = 7$ = smallest prime that does not divide

$\gcd(e, (p-1)(q-1)) = 1$ solve for e

$d = 343$ - used pulveriser

$(p-1 \cdot q-1, e)$

inverse of e mod $((p-1)(q-1))$

Fermat's Little Theorem

(d, n) is secret key (d,n)

(e, n) is public key, write on board

Message: 235

Can't combine - too big

adding pattern

Message: 3 to table 13 (their public key) $(5, 221)$

← Their public key

$m^* = (\text{rem } 3^5, ~~527~~ 221)$

$m^* = 13$

②

Message: 2 to 10 → public key (7, 209)

rem$(2^7, 209)$

↑ from their public key

82

~~Message "2" to 10 ...~~

Decrypting "99" from 6 → public key (5, 377)

rem$(99^{343}, 527)$ ← our private keys

Fast exponentiation mod 527

Or wolfram alpha

6 ✓ They say correct

Decrypting "369" from 13 → (5, 221)

rem$(369^{343}, 527)$

Can do fast exponentiation

7 ✓

Can't receive same message as ~~our pub~~ their n (public)

(3) Or the factors of it (since product of two primes)

Try to crack 13's secret key 221 = n

$p, q = 13, 7$

But d follows from this easily

~~gcd(e, 12·6) = 1~~ e is given, duh

~~gcd(e, 72) = 1~~

Then ~~first~~ find inverse for d

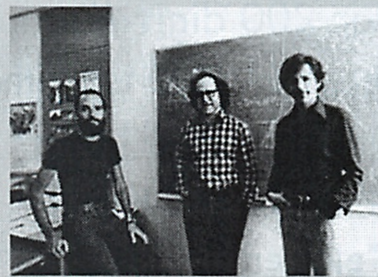
└ Fermat's Little Theorem

3/7

Mathematics for Computer Science
MIT 6.042J/18.062J

RSA encryption

RSA Public Key Encryption



Beforehand

receiver generates primes p, q

$n ::= p \cdot q$

selects e rel. prime to (p-1)(q-1)

(e, n) ::= public key, publishes it

finds d, inverse mod (p-1)(q-1) of e

d is secret key, keeps hidden

RSA

Encoding message $m \in [1,n)$

send $m^* ::= \text{rem}(m^e, n)$

Decoding m^*:

receiver computes

$\text{rem}((m^*)^d, n) = m$

Receiver's abilities

find two large primes p, q
- ok because: lots of primes
- fast test for primality

find e rel. prime to (p-1)(q-1)
- ok: lots of rel. prime nums
- gcd easy to compute

find (mod (p-1)(q-1)) inverse of e
- easy using Pulverizer or Euler

lots of primes

Prime Number Thm:

$\pi(n) ::= |\text{primes} \le n|$

$\sim n/\ln n$ (deep thm)

$\pi(n) > n/4 \log n$

Chebyshev's bound

"elementary" proof

lots of primes

so for 200 digit #'s,
at least 1/1000 is prime

$\pi(n) > n/4 \log n$

Chebyshev's bound
"elementary" proof

test if n is prime

check if

$$\text{rem}(a^{n-1}, n) = 1$$

if fails, not prime (Fermat)
choose random a in [1,n).
if not prime, Pr(fails)>1/2
(with rare exceptions)

Why does this work?

follows easily from Euler's Theorem when m has inverse mod n

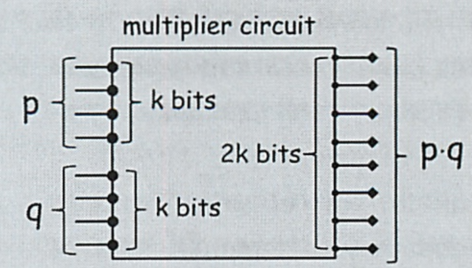
Why does this work?

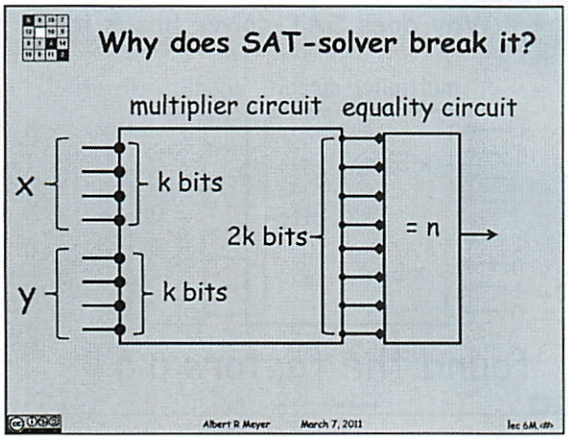
actually works for all m ... explained in Class Problem 2

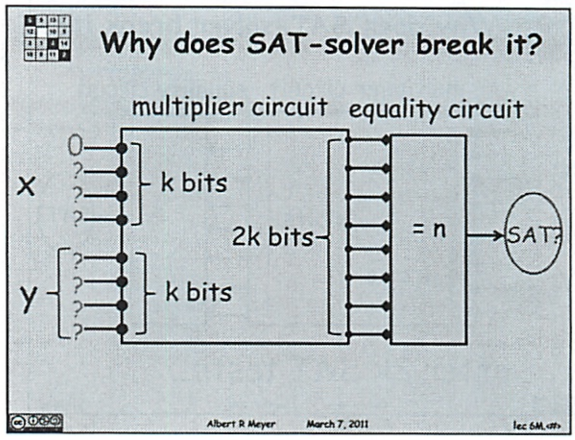
Why is it secure?

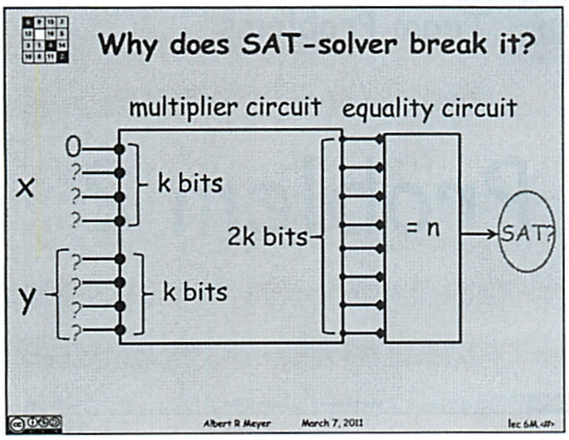
- easy to break *if* can factor n (find d same way receiver did)
- conversely, from d can factor n (but factoring appears hard so finding d must also be hard)
- RSA has withstood 30 years of attacks

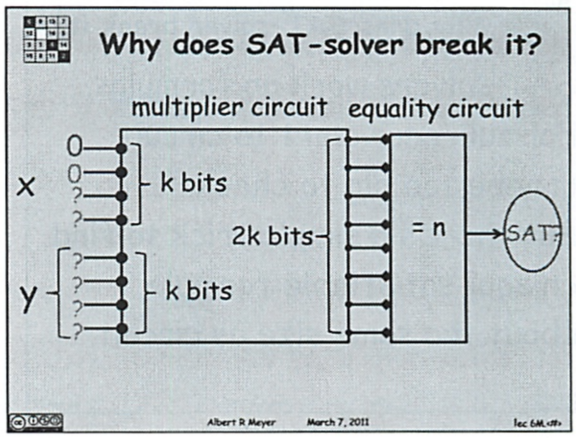
Why does SAT-solver break it?

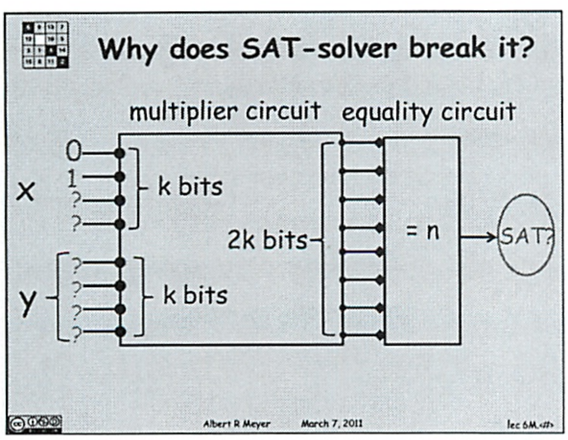


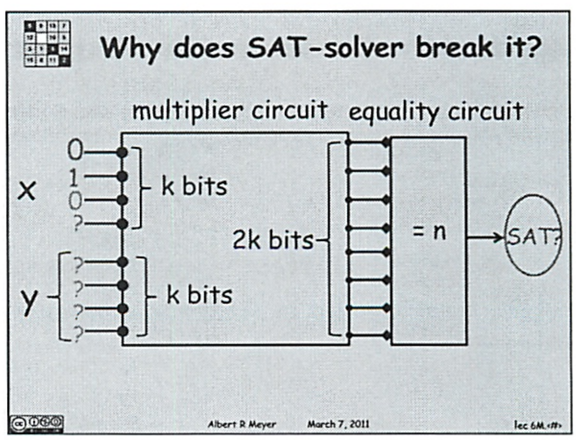

Why does SAT-solver break it?
multiplier circuit
equality circuit
x
k bits
y
k bits
2k bits
= n
Albert R Meyer
March 7, 2011


Why does SAT-solver break it?
multiplier circuit
equality circuit
x
0
?
?
?
k bits
y
?
?
?
?
k bits
2k bits
= n
SAT?
Albert R Meyer
March 7, 2011

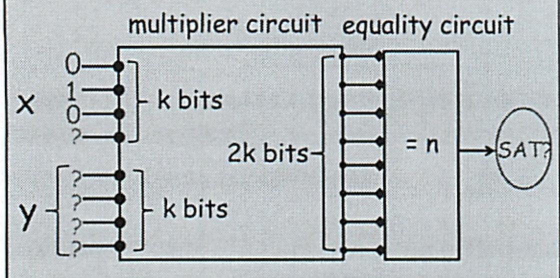

Why does SAT-solver break it?
multiplier circuit
equality circuit
x
0
?
?
?
k bits
y
?
?
?
?
k bits
2k bits
= n
SAT?
Albert R Meyer
March 7, 2011


Why does SAT-solver break it?
multiplier circuit
equality circuit
x
0
0
?
?
k bits
y
?
?
?
?
k bits
2k bits
= n
SAT?
Albert R Meyer
March 7, 2011


Why does SAT-solver break it?
multiplier circuit
equality circuit
x
0
1
?
?
k bits
y
?
?
?
?
k bits
2k bits
= n
SAT?
Albert R Meyer
March 7, 2011

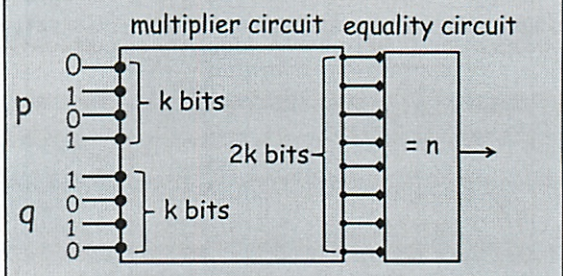

Why does SAT-solver break it?
multiplier circuit
equality circuit
x
0
1
0
?
k bits
y
?
?
?
?
k bits
2k bits
= n
SAT?
Albert R Meyer
March 7, 2011

Why does SAT-solver break it?



after 2k SAT tests…

Why does SAT-solver break it?



found the factors p,q !

Why does SAT-solver break it?

SAT-solvers work on formulas.
Formula equivalent to circuit may be too big to check.
But there's a simple trick to find an equi-satisfiable formula about the same size as circuit.

Team Problems

Problem 2

6.042 RSA

Anyone who has your ~~priv~~ public key can send you their message

Some people can play mental chess

Can you play mental poker
- who deals?
- Can do w/ public key

Paradoxial? – can compute some functions easily
but hard to reverse
like factoring

NSA tried to block them from publishing

Do 1 time connection via RSA to share a symetric cypher

Beforehand

- need pulvierser — Eulers ϕ
  - fast exponentiation
  - gcd
  - Fermat's ~~the~~ Little Theory

need 300 digit primes

(not ~~ap~~ copying algorithm – on sheet)

$n \sim 600$ digits

(2)

e is ϕ

- try at random until you find one, reditively ~~quickly~~ quickly
- rel. prime to $(p-1)(q-1)$

Publish (e, n).

Find d → inverse mod $(p-1)(q-1)$ of e

w/ pulverizer

d is secret key

Message must be broken up, so $m \in [1, n)$

- Send block by block

$m^* = rem(m^e, n)$

Decode ~~m~~ m^*

$m = rem((m^*)^d, n)$

Recievers's abilities

- finding p, q 2 large primes
  - one in every 1000 or so is prime
  - quick test for primality
- finding e rel prime to $(p-1)(q-1)$
  - lots of rel prime #
  - pick randomly ~1000 tries
  - easy to recognize ~~it~~

③

notes - checking for prime

Prime # Theorm

- Complex proof
- ~~pret~~ deep theorm

$\pi(n) ::= |\text{primes} \leq n|$

$\sim \frac{n}{\ln(n)}$ is prime

- limit is tachy - need error paths
  - how fast does it happen?

$\pi(n) > \frac{n}{4 \log n}$

- Chebcher's Bound
  - who would have thought ~~about~~ of this?

How to test if n is prime?

Fermat's Theorm

$rem(a^{n-1}, n) = 1$

$a^{n-1} \equiv 1 \mod (n)$

pick an a if not random
↳ fails → not prime
→ passes all the time → prime

(4)

can pass the test and <u>not</u> be prime

P(fails) for random a [1, n) ~~the~~ over $\frac{1}{2}$ times

L rare exception Carmon #
  ↑ or some name

So P(passes test | not prime) ≈ $\frac{1}{200}$

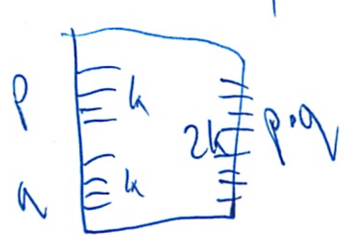
<u>Works for all m?</u> - see problem

Why is it secure?

- easy to break if can factor n
  - find d same way reciever did
- Converstey, from d we can factor n
- but factoring appears hard, so finding d must also be hard
- Theoritical security not that strong
- but withstood 30+ years of attacks

<u>SAT - Solver</u>

- Satisfiability



(5)

If could find SAT - could multiple quickly to factor

Multiplier circuit not that hard

So attach an equality test ~~thm~~ for n, set = to 1 to show equality

Then try first input to 0

Is it possible to fill in the other digits so the product = 1?

If sat, then set first bit to 0, move to 2nd bit and try 0

Run SAT again, if does not works, set 2nd bit to 1

Repeat

Then have binary representation of 1

But SAT solvers should work on formulas, ~~an~~ not circuits

- does not matter

Simpler trick equi-sat

In class prombles after

3/7

2. Congruent mod 10 means last digit is the same

Solutions to In-Class Problems Week 6, Mon.

Problem 1.
Let's try out RSA! There is a complete description of the algorithm in the text box. You'll probably need extra paper. **Check your work carefully!**

(a) As a team, go through the **beforehand** steps.

- Choose primes p and q to be relatively small, say in the range 10-40. In practice, p and q might contain several hundred digits, but small numbers are easier to handle with pencil and paper.
- Try $e = 3, 5, 7, \ldots$ until you find something that works. Use Euclid's algorithm to compute the gcd.
- Find d (using the Pulverizer—see appendix for a reminder on how the Pulverizer works—or Euler's Theorem).

When you're done, put your public key on the board. This lets another team send you a message.

(b) Now send an encrypted message to another team using their public key. Select your message m from the codebook below:

- 2 = Greetings and salutations!
- 3 = Yo, wassup?
- 4 = You guys are slow!
- 5 = All your base are belong to us.
- 6 = Someone on *our* team thinks someone on *your* team is kinda cute.
- 7 = You *are* the weakest link. Goodbye.

(c) Decrypt the message sent to you and verify that you received what the other team sent!

Problem 2.
A critical fact about RSA is, of course, that decrypting an encrypted message always gives back the original message! That is, that $\text{rem}((m^d)^e, pq) = m$. This will follow from something slightly more general:

Lemma 2.1. *Let n be a product of distinct primes and $a \equiv 1 \pmod{\phi(n)}$ for some nonnegative integer, a. Then*

$$m^a \equiv m \pmod{n}. \tag{1}$$

(a) Explain why Lemma 2.1 implies that k and k^5 have the same last digit. For example:

$$\underline{2}^5 = 3\underline{2} \qquad 7\underline{9}^5 = 307705639\underline{9}$$

Hint: What is $\phi(10)$?

Solution. Two nonnegative integers have the same last digit iff they are $\equiv \pmod{10}$. Now $\phi(10) = \phi(2)\phi(5) = 4$ and $5 \equiv 1 \pmod 4$, so by Lemma 2.1,

$$k^5 \equiv k \pmod{10}.$$

■

(b) Explain why Lemma 2.1 implies that the original message, m, equals $\text{rem}((m^e)^d, pq)$.

Solution. To apply Lemma 2.1 to RSA, note that the first condition of the Lemma is that n be a product of primes. In RSA, $n = pq$ so this condition holds.

For $n = pq$, we have from from Lemma 8.7.5 or the more general the Theorem 8.7.6 that $\phi(n) = (p-1)(q-1)$. So when d and e are chosen according to RSA, $de \equiv 1 \pmod{\phi(n)}$. So $a ::= de$ satisfies the second condition of the Lemma.

Now, from equation (1) with $n = pq$ and $a = de$, we have

$$(m^e)^d = m^{de} \equiv m \pmod{pq}.$$

Hence,

$$\text{rem}((m^e)^d, pq) = \text{rem}(m, pq),$$

but $\text{rem}(m, pq) = m$, since $0 \le m < pq$. ■

(c) Prove that if p is prime, then

$$m^a \equiv m \pmod p \tag{2}$$

for all nonnegative integers $a \equiv 1 \pmod{p-1}$.

Solution. If $p \mid m$, then equation (2) holds since both sides of the congruence are $\equiv 0 \pmod p$.

So assume p does not divide m. Now if $a \equiv 1 \pmod{p-1}$, then $a = 1 + (p-1)k$ for some k, so

$$\begin{aligned}
m^a &= m^{1+(p-1)k} \\
&= m \cdot \left(m^{p-1}\right)^k \\
&\equiv m \cdot (1)^k \pmod p && \text{(by Fermat's Little Thm.)} \\
&\equiv m \pmod p.
\end{aligned}$$

■

(d) Prove that if $a \equiv b \pmod{p_i}$ for distinct primes $p_1, p_2, \ldots, p_n$, then $a \equiv b \pmod{p_1 p_1 \cdots p_n}$.

Solution. By definition of congruence, $a \equiv b \pmod k$ iff $k \mid (a-b)$. So if $a \equiv b \pmod{p_i}$ for each p_i, then $p_i \mid (a-b)$ for each p_i. By the Unique Factorization Theorem 8.3.1, the product of the p_i's must also divide $(a-b)$, which means that $a \equiv b \pmod{p_1 p_1 \cdots p_n}$. ■

(e) Combine the previous parts to complete the proof of Lemma 2.1.

Solution. Suppose n is a product of distinct primes, $p_1 p_2 \cdots p_k$. Then from the formulas for the Euler function, ϕ, we have

$$\phi(n) = (p_1 - 1)(p_2 - 1) \cdots (p_k - 1).$$

Now suppose $a \equiv 1 \pmod{\phi(n)}$, that is, a is 1 plus a multiple of $\phi(n)$, so it is also 1 plus a multiple of $p_i - 1$. That is,

$$a \equiv 1 \pmod{p_i - 1}.$$

Hence, by part (c),

$$m^a \equiv m \pmod{p_i}$$

for all m. Since this holds for all factors, p_i, of n, we conclude from part (d) that

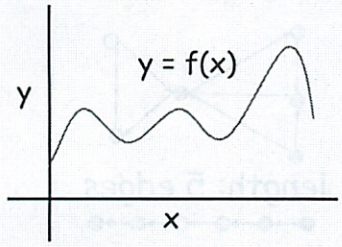
$$m^a \equiv m \pmod{n},$$

which proves Lemma 2.1. ■

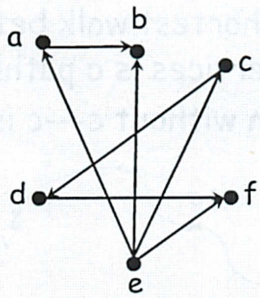
Mathematics for Computer Science
MIT 6.042J/18.062J

Directed Graphs

Normal Person's Graph



Computer Scientist's Graph

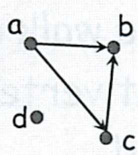


Digraphs

- a set, V, of vertices
- a set, $E \subseteq V \times V$ of directed edges

$(v,w) \in E$ notation: $v \rightarrow w$

Relations and Graphs



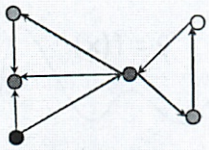
V= {a,b,c,d}
E = {(a,b), (a,c), (c,b)}

Digraphs

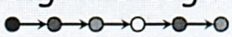
Formally, a digraph with vertices V is *the same* as a binary relation on V.

Walks & Paths

Walk: follow successive edges



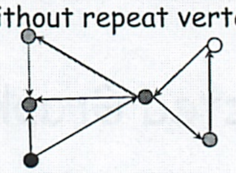
length: 5 edges



(not 6 vertices)

Walks & Paths

Path: walk thru vertices without repeat vertex



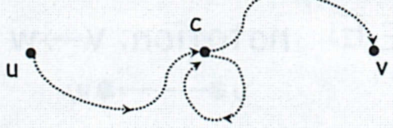
length: 4 edges

Walks & Paths

Lemma:

The shortest walk between two vertices is a path!

Proof: (by contradiction) suppose path from u to v crossed itself:

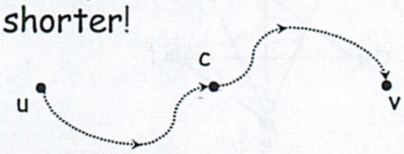


Walks & Paths

Lemma:

The shortest walk between two vertices is a path!

then path without c---c is shorter!



Walks & Paths

Digraph G defines path relation G^+

u G^+ v iff $\exists$path u to v (lnth > 0)

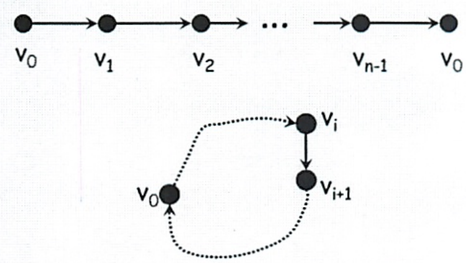
(the positive path relation)

Cycles

A cycle is a walk whose only repeat vertex is its start & end.

(a single vertex is a length 0 cycle)

Cycles



Directed Acyclic Graph

DAG

has no positive length cycle

Directed Acyclic Graph

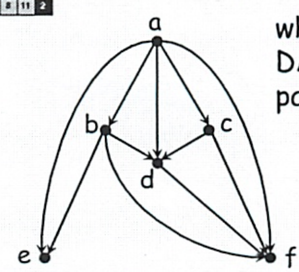
examples: DAG

< relation on integers

⊂ relation on sets

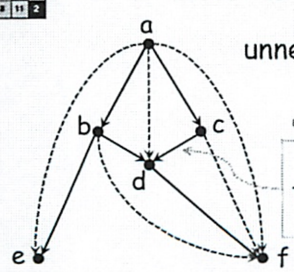
prerequisite on classes

DAG path relation



what is *smallest* DAG with same path relation?

Covering Edges



unneeded edges

covering edges

e.g. any path from c to d must use c→d

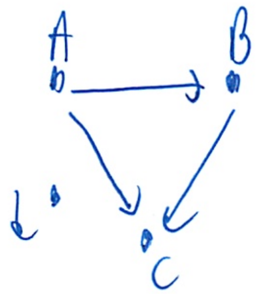
Problems

1 – 3

What is smallest called?
↳ ? w/ the covering edges

6.042 Directed Graphs

3/9



a set V of verticies

a set $E \subseteq V \cdot V$ of directed edges

notation $V \to W$

$V \longrightarrow \bullet W$

$V = \{a, b, c, d\}$

$E = \{\langle a,b \rangle, \langle a,c \rangle, \langle b,c \rangle\}$

Same as binary relation where domain + codomain are same

Walks and Paths

Follow successive edges to make a ~~"path"~~ "walk"

Some people break stuff down to "verticies" and "edges"

Can cross itself by going through same vertex

lenght of walk = # of edges
(# verticies − 1)

1 vertex = degenerative path (lenght 0)

②

Can get stuck if no edges out

path - can't repeat

The shortest walk b/w 2 vertices is a path

Proof by condiction

<- just drop the loop

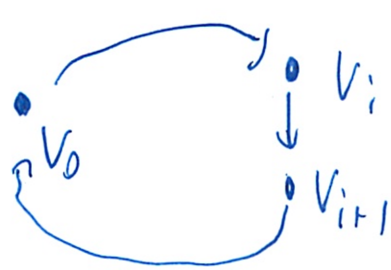
Diagraph represents path relation G^+

$u \, G^+ \, v$ $\exists$ path u to v

↳ must also be + lenght path

Cycles - positive lenght ~~path~~ walk whose only vertex that repeats is start + end

-Cycle can be of lenght 0



(3)

DAG = <u>Directed Acyclic Graph</u>

has no positive lenght cycle

Problem: how to find DAG w/ smallest # of relations
w/ same path relations

Cut out unneeded paths

But have to ~~leave~~ leave all the points connected that were connected

What is the best way to get minimal sized one

Will be min-unique solution w/ DAG

All remaining pts are "covering edge"

- must be kept
- so ~~we~~ whats left is unique

In-Class Problems Week 6, Wed.

Problem 1.
In a round-robin tournament, every two distinct players play against each other just once. For a round-robin tournament with with no tied games, a record of who beat whom can be described with a *tournament digraph*, where the vertices correspond to players and there is an edge $\langle x \to y \rangle$ iff x beat y in their game.

A *ranking* is a path that includes all the players.

(a) Give an example of a tournament digraph with more than one ranking.

(b) Prove that if a tournament digraph is a DAG, then it has at most one ranking. *Hint:* Prove that the elements below u in any ranking are uniquely determined.

(c) Prove that every finite tournament digraph has a ranking.

(d) Give an example of a tournament with a countably infinite number of players, $p_0, p_1, \dots$ that has no ranking.

Hint: $\mathbb{Q}$.

Problem 2.
If a and b are distinct nodes of a digraph, then a is said to *cover* b if there is an edge from a to b and every path from a to b traverses this edge. If a covers b, the edge from a to b is called a *covering edge*.

(a) What are the covering edges in the DAG in Figure 1?

(b) Let covering (D) be the subgraph of D consisting of only the covering edges. Suppose D is a finite DAG. Explain why covering (D) has the same positive path relation as D.

Hint: Consider *longest* paths between a pair of vertices.

(c) Show that if two *DAG*'s have the same positive path relation, then they have the same set of covering edges.

(d) Conclude that covering (D) is the *unique* DAG with the smallest number of edges among all digraphs with the same positive path relation as D.

The following examples show that the above results don't work in general for digraphs with cycles.

(e) Describe two graphs with vertices $\{1, 2\}$ which have the same set of covering edges, but not the same positive path relation (*Hint:* Self-loops.)

(f) (i) The *complete digraph* without self-loops on vertices $1, 2, 3$ has edges between every two distinct vertices. What are its covering edges?

(ii) What are the covering edges of the graph with vertices $1, 2, 3$ and edges $\langle 1 \to 2 \rangle$, $\langle 2 \to 3 \rangle$, $\langle 3 \to 1 \rangle$?

(iii) What about their positive path relations?

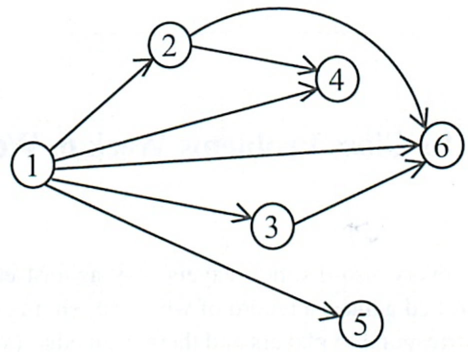


Figure 1 DAG with edges not needed in paths

Problem 3.
A 3-bit string is a string made up of 3 characters, each a 0 or a 1. Suppose you'd like to write out, in one string, all eight of the 3-bit strings in any convenient order. For example, if you wrote out the 3-bit strings in the usual order starting with 000 001 010..., you could concatenate them together to get a length $3 \cdot 8 = 24$ string that started 000001010....

But you can get a shorter string containing all eight 3-bit strings by starting with 00010.... Now 000 is present as bits 1 through 3, and 001 is present as bits 2 through 4, and 010 is present as bits 3 through 5, ...

(a) Say a string *3-good* if it contains every 3-bit string as 3 consecutive bits somewhere in it. Find a 3-good string of length 10, and explain see why this is the minimum length for any string that is 3-good.

(b) Explain how any walk that includes every edge in the graph shown in Figure 2 determines a string that is 3-good. Find the walk in this graph that determines your good 3-good string from part (a).

(c) Explain why a path in the graph of Figure 2 that includes every every edge *exactly once* provides a minimum length 3-good string.

(d) The situation above generalizes to $k \geq 2$. Namely, there is a digraph, B_k, such that $V(B_k) ::= \{0, 1\}^k$, and any walk through B_k that contains every edge exactly once determines a minimum length $(k+1)$-good bit-string. What is this minimum length?

Define the transitions of B_k. Verify that the in-degree and out-degree of every vertex is even, and that there is a positive path from any vertex to any other vertex (including itself) of length at most k.[1]

[1]Problem 9.6 shows that if the in-degree of every vertex of a digraph is equal to its out-degree, and there are paths between any two vertices, then there is a closed walk that includes every edge exactly once. So the graph B_k implies that there always is a length-$2^{k+1} + k$ bit-string in which every length-$(k+1)$ bit-string appears as a substring. Such strings are known as *de Bruijn sequences*.

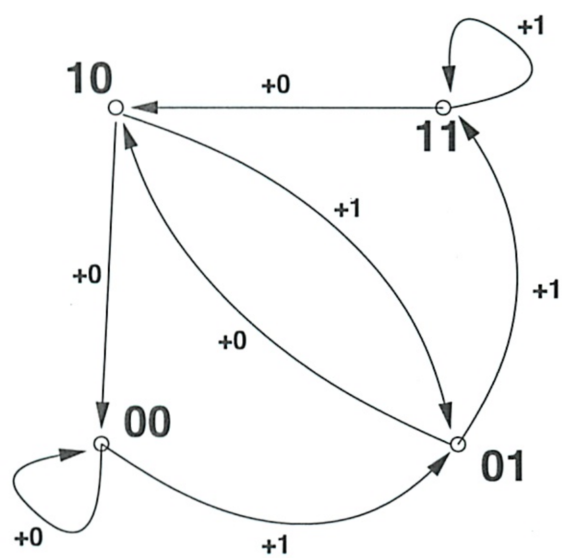


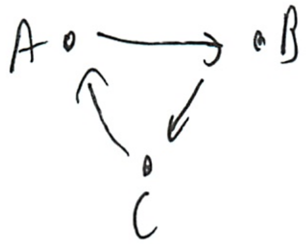
Figure 2 The 2-bit graph.

In Class Problems

3/9

1a. say 3 players A B C

A → B
B → C
C → A ~~then~~



So tom tied

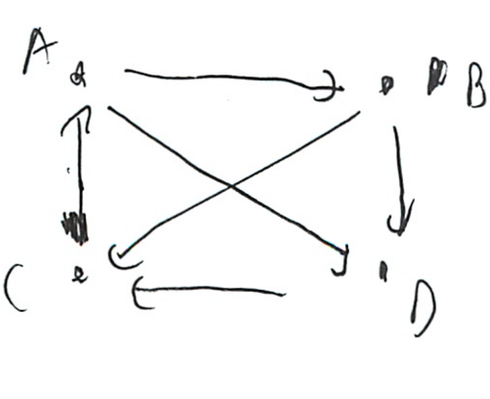
b) Prove that is DAG if at most 1 ranking

ranking = path that includes all players

So means only 1 unique path through them

Try 4 random

A → B
A ← C
A → D
B ← C
B → D
C ← D



So is it a cycle

No - not a ~~cycle~~ path
So not unique ranking

②

b) But how do you prove that?
What do you write?
What is a_i? Hasn't been defined

c) But it does not! It will go through points multiple times.

d) Give example w/ countably ∞ players w/ no ranking
No way to include all players if ∞?

2.

1 → 2 → 4 is not covering 1 → 4

Oh right the goal is for DAG to have <u>only</u> covering edges

b) Aren't these the same things?
No - see fig1 DAG ≠ only covering edges
DAG = no cycles
Can only remove edges if another path to that vertex

c) There is only 1 set of covering edges - it is a unique solution

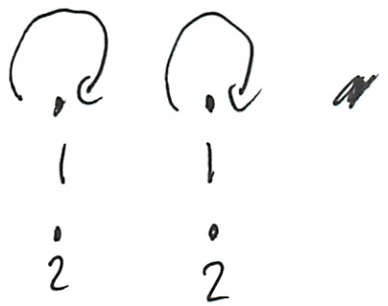
(3)

d)?: By def.

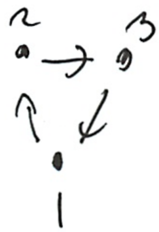
Or proove def

e) Describe 2 graphs vertices {1,2} w/ same set of covering edges

- self loop



f)



What does complete mean?

What is def covering edges again

ii) Same as I had above

Need to think more slowly/carefully about

(9)

3. 3 bit string

000
001
011
101
110
100
010
000

Can cat 8 together for 24 string
But can get shorter ~~bc~~ string starting w/ 00010
000 = bits 1→3
001 = bits 2→4
010 = bits 3→5

a) Say string 3-good if contains every 3 bit string in there
- with length 10

Oh I see what they mean above - each window

000 10 11 0 111   ℓ 11 char ~~strings~~

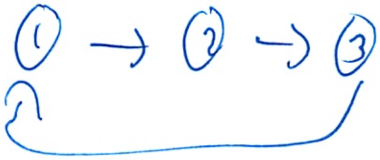
- 000
- 001
- 010
- 011 ←
- 110
- 101   duplicate
- 011 ←
- 111   ≈ 1 missing

not unique
- could reverse for one

Walk through fig 3 at least once
- multiple pts ok
- not multiple paths

(5)

1a)



Not a DAG

more like a DCG

1b) A DAG would have 1 person who loses to everyone else

<u>Proof</u> If its a DAG $\exists n$ ~~(maybe more than 1)~~ who loses to everybone

Remove the biggest loser - graph still is a DAG

Still $\exists! n$ who loses to everybody.

So by Induction

We get reverse ordering of the ranking

Can't ignore trivial cases of 0

Lose - define → That is tornirent

Better describe case w/ 1 game

Have to prove if each person won once ~~its a DAG~~

- hardest part
- Shows that not a DAG
- You are not prooving

This is more of a proof of C

(6)

our
3 board

a) 000 1011100

- Must have 3 el to contain a 3 bit string
- must gain at least one other in addition to base 3

$3+7=10$

b) Every edge defines a unique 3-bit string

00, +0 = 000

00, +1 = 001 etc

A string produced by a walk on the graph will Contain all of the 3 bit strings defined by the transvered edges

So if all edges are transversed - string 3 good

000 1011100

00 → 0 → 1 → 0 → 1 → 1 → 1 → 0 → 0

c) A string produced by a walk that transverses each edge exactly once contains 2 bits from initial state + 1 bit for each of the 8 edges

$8+2=10$ = min length

(7)

d) min length $k + 2^{k+1}$

2 transitions from each vertex

~~Bit~~ shift bit string left, append 1 or 0

2 transitions into each vertex

Shift bit string right, prepend 1 or 0

I redo ~~board~~ our board

Proof If a DAG ∃n who only loses

if you walk the graph you will go node by done + never return to node already visited (since no cycles allowed)

∴ You will end up at a node and can not go ~~the~~ anywhere else - loser

If only 2 nodes, one wins, one loses

So only one will lose

Biggest Loser is last el

Remove, repeat

That is bottom up printing of ranking

Only one biggest loser at each stage

Solutions to In-Class Problems Week 6, Wed.

Problem 1.
In a round-robin tournament, every two distinct players play against each other just once. For a round-robin tournament with with no tied games, a record of who beat whom can be described with a *tournament digraph*, where the vertices correspond to players and there is an edge $\langle x \to y \rangle$ iff x beat y in their game.

A *ranking* is a path that includes all the players.

(a) Give an example of a tournament digraph with more than one ranking.

Solution. Let $n = 3$ with edges $\langle u \to v \rangle$, $\langle v \to w \rangle$ and $\langle w \to u \rangle$. Then both u, v, w and v, w, u are rankings. ■

(b) Prove that if a tournament digraph is a DAG, then it has at most one ranking.

Solution. Suppose for contradiction that there are two rankings for the graph. Since the rankings differ, there must be two players $u \neq v$ such that u ranks higher than v in one ranking and lower than v in the other ranking. So one ranking gives a path from u to v and the other ranking gives a path from v to u.

Merging these paths gives a closed *walk* from u to u that goes through v. From this we would like to conclude that there is a positive length cycle from u to u. This would contradict the fact that the graph is a DAG, and so would complete the proof.

But having a closed walk of from u to u that goes through v does not by itself imply that there is a *cycle* from from u to u that goes through v. In fact, in general there may not be such a cycle —an example of this appears at the end of this solution.

Now there are two ways to close this loophole. One is to observe that

Lemma. *The shortest positive length closed walk through a vertex is a cycle.*

Since a walk from from u to u that goes through $v \neq u$ must have positive length, this Lemma implies there is a positive length cycle from u to u (somewhere, not necessarily though v), contradicting the fact the graph is a DAG and so completing the overall proof.

All that remains is proving the Lemma, and the proof of the Lemma is essentially the same as for Theorem 9.2.4 that a shortest walk is a path.

of the Lemma. Suppose $\mathbf{w}$ is a minimum positive length walk from u to u. We claim $\mathbf{w}$ is a cycle.

To prove the claim, suppose to the contrary that $\mathbf{w}$ is not a cycle.

case (u occurs more than two times in $\mathbf{w}$): This means that

$$\mathbf{w} = \mathbf{e}\,\widehat{u}\,\mathbf{f}$$

where both $\mathbf{e}$ and $\mathbf{f}$ have positive length. Then $\mathbf{e}$ is a shorter positive length walk from u to u, contradicting the minimality of $\mathbf{w}$.

case (some vertex $x \neq u$ occurs twice in $\mathbf{w}$): Then

$$\mathbf{w} = \mathbf{e}\,\widehat{x}\,\mathbf{f}\,\widehat{x}\,\mathbf{g}$$

for some positive length walks $\mathbf{e}, \mathbf{f}, \mathbf{g}$. But then "deleting" $\mathbf{f}$ yields a strictly shorter walk, namely

$$\mathbf{e}\,\widehat{x}\,\mathbf{g}$$

is a shorter walk from u to u, again contradicting the minimality of $\mathbf{w}$. ■

The second way out of the loophole is to observe that in a tournament graph, there must be an *edge* in one direction or the other between u and v. So say the edge is $\langle u \rightarrow v \rangle$. Then this edge merged with the path from v to u will be a cycle (think about why).

By the way, another workable approach to this problem is by induction on the number of vertices, which we omit.

Example.

$$V ::= \{u, v, w, x\},$$
$$E ::= \{\langle u \rightarrow w \rangle, \langle w \rightarrow x \rangle, \langle x \rightarrow u \rangle, \langle v \rightarrow w \rangle, \langle x \rightarrow v \rangle\},$$

there is a path

$$u\ \langle u \rightarrow w \rangle\ w\ \langle w \rightarrow x \rangle\ x\ \langle x \rightarrow v \rangle$$

from u to v, and a path

$$v\ \langle v \rightarrow w \rangle\ w\ \langle w \rightarrow x \rangle\ x\ \langle x \rightarrow u \rangle\ u$$

from v to u, but it is easy to see that there is no *cycle* from u to u *that contains* v. (The sole edge out of u goes to w, and the sole edge out of v likewise goes to w, so any walk from u to u that goes through v must go through w at least twice and therefore won't be a cycle.

■

(c) Prove that every finite tournament digraph has a ranking.

Solution. By induction on n with induction hypothesis

$$P(n) ::= \text{every tournament digraph with } n \text{ vertices has a ranking.}$$

base case $n = 1$: Trivial.

inductive step: Let G be a tournament digraph with $n + 1$ vertices. Remove one vertex, v, to obtain the subgraph, H, with the n remaining vertices. Clearly, H is also a tournament digraph, so by induction hypothesis it has a ranking. Now if the last player in this H-ranking beat player v, then v can be added at the end to form a ranking in G. On the other hand, if v beat the last player in the H-ranking, then there will (by WOP) be a first player in the H-ranking that v beats. Inserting v just before that first player gives a ranking for G. Since G was an arbitrary $n + 1$ vertex tournament graph, we conclude that $P(n + 1)$ holds, which completes the proof. ■

(d) Give an example of a tournament with a countably infinite number of players, $p_0, p_1, \ldots$ that has no ranking.

Hint: $\mathbb{Q}$.

Solution. The rationals, $\mathbb{Q}$, are a countable set, and specifying that r beats s precisely when $r > s$ defines a tournament graph with $\mathbb{Q}$ as the set of players.

Now in any tournament graph, vertex u can come before vertex u in some ranking only if there is a path from u to v. This implies that if $r > s$, then r must come before s in *any* ranking of $\mathbb{Q}$.

So suppose there was a ranking of $\mathbb{Q}$ and $\langle r \to s \rangle$ was an edge on the path. This implies that $r > s$. Now let t be any rational such that $r > t > s$. Now in a ranking, t must come before r or after s, which implies $t > r$ or $s > t$, a contradicting the choice of t. SO there cannot be a ranking of the $\mathbb{Q}$ tournament. ■

Problem 2.
If a and b are distinct nodes of a digraph, then a is said to *cover* b if there is an edge from a to b and every path from a to b traverses this edge. If a covers b, the edge from a to b is called a *covering edge*.

(a) What are the covering edges in the DAG in Figure 1?

Solution. TBA ■

(b) Let covering (D) be the subgraph of D consisting of only the covering edges. Suppose D is a finite DAG. Explain why covering (D) has the same positive path relation as D.

Hint: Consider *longest* paths between a pair of vertices.

Solution. What we need to show is that if there is a path in D between vertices $a \neq b$, then there is a path consisting only of covering edges from a to b. But since D is a finite DAG, there must be a *longest* path from a to b. Now every edge on this path must be a covering edge or it could be replaced by a path of length 2 or more, yielding a longer path from a to b. ■

(c) Show that if two *DAG*'s have the same positive path relation, then they have the same set of covering edges.

Solution. *Proof.* Suppose C and D are DAG's with the same positive path relation and that $\langle a \to b \rangle$ is a covering edge of C. We want to show that $\langle a \to b \rangle$ must also be a covering edge of D.

Since $\langle a \to b \rangle$ itself defines a (length one) positive length path in C, there must be a positive length path in D from a to b. If this positive length path in D is of length greater than one, then the path must consist of a positive length path from a to c followed by a positive length path from c to b for some vertex, c. Also, since D is a DAG, c cannot be a or b.

This means there must also be positive length paths in C from a to c and from c to b, and neither of these paths can traverse $\langle a \to b \rangle$ or there would be a cycle. Hence the path from a to c to b is a path in C that does not traverse $\langle a \to b \rangle$, contradicting the fact that $\langle a \to b \rangle$ is a covering edge of C.

In sum, there is a length one path from a to b in D, namely $\langle a \to b \rangle$, and this is the *only* path from a to b in D, which proves that $\langle a \to b \rangle$ is a covering edge in D. ■

(d) Conclude that covering (D) is the *unique* DAG with the smallest number of edges among all digraphs with the same positive path relation as D.

Solution. By part (c), any DAG with the same positive path relation as D must contain all the edges of covering (D). By part (b), covering (D) has this same positive path relation. It follows immediately that covering (D) is the unique minimum-size DAG with the same positive path relation as D. ■

The following examples show that the above results don't work in general for digraphs with cycles.

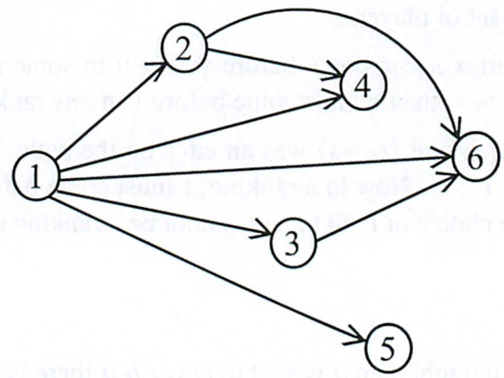


Figure 1 DAG with edges not needed in paths

(e) Describe two graphs with vertices $\{1, 2\}$ which have the same set of covering edges, but not the same positive path relation (*Hint:* Self-loops.)

Solution. Let one graph have edges $\{(1, 2), (1, 1)\}$ and the other $\{(1, 2), (2, 2)\}$. They have the same set of covering edges, namely, $(1, 2)$. But in the second there is a positive length path from 2 to 2, namely a path of length one but there is no positive length path from 2 to 2 in the first graph. ■

(f) (i) The *complete digraph* without self-loops on vertices 1, 2, 3 has edges between every two distinct vertices. What are its covering edges?

(ii) What are the covering edges of the graph with vertices 1, 2, 3 and edges $\langle 1 \to 2 \rangle$, $\langle 2 \to 3 \rangle$, $\langle 3 \to 1 \rangle$?

(iii) What about their positive path relations?

Solution. (i) There are no covering edges, since there is always a length two path from a to b that does not use the edge $\langle a \to b \rangle$.

(ii) All three edges are the covering edges.

(iii) They have the same positive path relation, namely, each vertex is connected to all the vertices, including itself, by positive length paths.

■

Problem 3.

A 3-bit string is a string made up of 3 characters, each a 0 or a 1. Suppose you'd like to write out, in one string, all eight of the 3-bit strings in any convenient order. For example, if you wrote out the 3-bit strings in the usual order starting with 000 001 010..., you could concatenate them together to get a length $3 \cdot 8 = 24$ string that started 000001010....

But you can get a shorter string containing all eight 3-bit strings by starting with 00010.... Now 000 is present as bits 1 through 3, and 001 is present as bits 2 through 4, and 010 is present as bits 3 through 5, ...

(a) Say a string *3-good* if it contains every 3-bit string as 3 consecutive bits somewhere in it. Find a 3-good string of length 10, and explain see why this is the minimum length for any string that is 3-good.

Solution. The string 0001110100 is a length 10 string that is 3-good. You can't do better: there must be two bits to start and each additional bit can yield at most one new 3-bit string. ■

(b) Explain how any walk that includes every edge in the graph shown in Figure 2 determines a string that is 3-good. Find the walk in this graph that determines your good 3-good string from part (a).

Solution. A string can be built up from any walk by starting with the k bits in the vertex at the start of the walk and successively adding the bit that labels the edge to the end of the string being built. If the walk includes every edge, then any string $b_1b_2b_3$ will appear as a substring when the edge $\langle b_1b_2 \to b_2b_3 \rangle$ appears in the walk.

In particular, the string 0001110100 is determined by the walk that goes through the following sequence of edges:

$$\langle 00 \to 00 \rangle \; \langle 00 \to 01 \rangle \; \langle 01 \to 11 \rangle \; \langle 11 \to 11 \rangle \; \langle 11 \to 10 \rangle \; \langle 10 \to 01 \rangle \; \langle 01 \to 10 \rangle \; \langle 10 \to 00 \rangle \,.$$

■

(c) Explain why a path in the graph of Figure 2 that includes every every edge *exactly once* provides a minimum length 3-good string.

Solution. Since there are 8 edges, the string determined by the walk will be of length 10, which is minimum possible as observed in part (a). Since the walk includes every edge, it will determine a 3-good string by part (b). ■

(d) The situation above generalizes to $k \geq 2$. Namely, there is a digraph, B_k, such that $V(B_k) ::= \{0, 1\}^k$, and any walk through B_k that contains every edge exactly once determines a minimum length $(k + 1)$-good bit-string. What is this minimum length?

Define the transitions of B_k. Verify that the in-degree and out-degree of every vertex is even, and that there is a positive path from any vertex to any other vertex (including itself) of length at most k.[1]

Solution. A string of length n has exactly $n - k$ locations where a length $k + 1$ subsequence can begin. Since there are 2^{k+1} length-$(k + 1)$ bit strings, the mininum length, n of any $(k + 1)$ good string must satisfy $n - k/ge2^{k+1}$, so the mininum length is at least $2^{k+1} + k$. This is exactly the length string that would be determined by a path containing all $2 \cdot 2^k$ edges in the graph B_k.

$$E(B_k) ::= \{\langle xa \to bx \rangle \mid x \in \{0, 1\}^{k-1} \text{ AND } a, b \in \{0, 1\}\}$$

If $y \in \{0, 1\}^k$, then $y = xa$ and $y = bz$ for unique strings $x, z \in \{0, 1\}^{k-1}$ and bits $a, b \in \{0, 1\}$. Then by definition of $E(B_k)$, there are exactly two edges out of y, one going to $0x$ and the other to $1x$, so $\text{outdeg}(y) = 2$. Likewise, there are only two edges into y, one from $z0$ and the other from $z1$, so $\text{outdeg}(y) = 2$.

To get from vertex $b_1b_2 \ldots b_k$ to $c_1c_2 \ldots c_k$ with a length k path, proceed as follows:

$$\begin{aligned} b_1b_2 \ldots b_k &\to c_kb_1b_2 \ldots b_{k-1} \to c_{k-1}c_kb_1b_2 \ldots b_{k-2} \\ &\to \cdots \to c_2c_3 \ldots c_kb_1 \to c_1c_2 \ldots c_k \end{aligned}$$

■

[1]Problem 9.7 shows that if the in-degree of every vertex of a digraph is equal to its out-degree, and there are paths between any two vertices, then there is a closed walk that includes every edge exactly once. So the graph B_k implies that there always is a length-$2^{k+1} + k$ bit-string in which every length-$(k + 1)$ bit-string appears as a substring. Such strings are known as *de Bruijn sequences*.

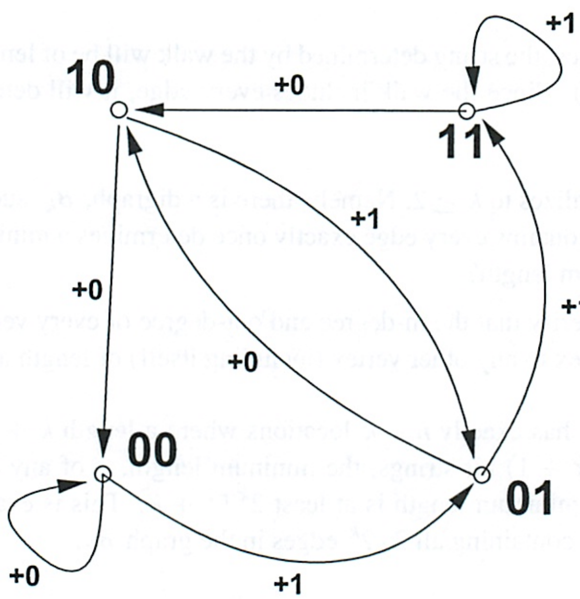


Figure 2 The 2-bit graph.

Mod Inverses Review

Inverses

$x \cdot x^{-1} = 1$

$3 \cdot \frac{1}{3} = 1$

$7 \cdot \_ \equiv 1 \pmod{5}$

↑ what can you fill in here

3

also all # congruent to 3 (mod 5)

- basically $3 + \_ \cdot 5$

↑ any integer (positive)

Pulverizer

Find inverse of k mod p ↑ prime

k must not be a multiple of p

So $\gcd(k, p) = 1$

So can have linear combo

$$sp + tk = 1$$

$$sp = 1 - tk$$

Which implies $p \mid (1 - tk)$ so tk must $\equiv 1 \pmod{p}$

Pulverizer is to find this linear combo

$$r = x - qy$$

anything that is multiple of gcd → $r = ax + by$

(2)

Let me try

$k = 3$

$p = 5$

So $\gcd(5,3)$:

$\gcd(3, \text{rem}(5,3))$ — $2 = 5 - 3 \cdot 1$

$\gcd(2, \text{rem}(3,2))$ — $1 = 3 - 2 \cdot 1$

$\gcd(1, \text{rem}(2,1))$ — $1 = 3 - (5 - 3 \cdot 1) \cdot 1$

$\gcd(1,0) = 1$ — $1 = 3 \cdot 2 - 5 \cdot 1$

done

(feel better doing w/o looking)

$a = -1$

$b = 2$ ← So answer's

test

$3 \cdot 2 = 1 \pmod 5$ ✓

Cancelable if $m_1 k = m_2 k$

Cancle ks → $m_1 = m_2$

* Can only cancle w/ mod prime

(3)

Fermats' Little Theorm

$$k^{p-1} = 1 \mod p$$

To find inverse

$$k^{p-2} \cdot k \equiv 1 \pmod p$$

So ~~take~~ for $k = 3$ $p = 5$

$rem(3^3, 5)$

Can do fast exponentiation

Set $x = a$

$y = 1$

$z = b$

loop:

- if $z = 0$: return y + terminate
- $r = rem(z, 2) \mod p$
- $z = quot(z, 2) \mod p$
- if $r = 1$ $y = xy \mod p$
- $x = x^2 \mod p$

Or could also do plain-jane exponentiation

$rem(27, 5) = 2$ ans!

Test $3 \cdot 2 \equiv 1 \mod 5$ ✓

(4)

Euler's Theorm $\phi(\ )$

∟ generalization Fermat's Little Theorm

\# of relatively prime #s

$\phi(p) = p-1$
↑ prime

$k^{\phi(n)} \equiv 1 \pmod{n}$
↑ all int rel. prime to n
↑ k is rel. prime to n

(✱) If k is rel prime to n

$k^{\phi(n)-1}$ = multiplicitive inverse k mod n

To compute $\phi(n)$

∟ need prime factorization of n
Pulverizer better if n is large

For $\phi(pq)$
↑↑ primes

$\phi(pq) = (p-1)(q-1)$

For $\phi(p^k) = p^k - p^{k-1}$
↑ prime

For $\phi(ab) = \phi(a)\,\phi(b)$
↑ rel prime to each other

(5)

So example $\phi(300)$

know prime factorization $300 = 2^2 \cdot 3 \cdot 5^2$

$= \phi(2^2 \cdot 3 \cdot 5^2)$

$= \phi(2^2) \cdot \phi(3) \cdot \phi(5^2)$

$= (2^2 - 2^1)(3^1 - 3^0)(5^2 - 5^1)$

$= 80$

Now test finding ~~prime~~ inverse

$k = 3$

$p = 5$

$3^{\phi(5) - 1}$

[$\phi(5) =$ ~~5A~~ $5 - 1 = 4$ since prime

$= 3^{4-1}$

$= 3^3$

$= 27$

I guess is one

27 (mod 5) = (2) bingo

↑ do

Tutor 6

6.1 Divisablity DAG

- divisibity relation on {1, ..., 12}
- upward path from a → b iff a | b
- If add 24, how many edges to added

1 2 3 4 6 8 12 24

So we are not removing duplicate relations?

But only need from top

not 1, 2, 3, 4, 6, 8 only

9 does not work so

So 3 8 12 = 3 ✗

8

DAGs represent partial orders economically

L < than total set ??

2

8 → 24
12 → 24

? so why not 3
- oh guess since
3 → 6 - it branches

②

TP2 Matrix representation of relations

$M_R = \begin{matrix} 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{matrix}$

Find ~~M~~ $M_{R^{-1}}$

- ? is this just inverse of matrix
flipping arrows

Use matlab

? gives

$\begin{matrix} -.5 & .5 & .5 \\ .5 & .5 & -.5 \\ .5 & -.5 & .5 \end{matrix}$ weird!

Or just T transpose ?

- same

Just do it manually

	a	b	c
a	0	1	1
b	1	1	0
c	1	0	1

← just write writing down ? what is that called ?

Oh is the same - Thats why I thouht it was broken

0

(3)

b) Complement (R)

Matlab

-not in

-determinant of A where all rows + columns of

M have been removed

{minor

Where is in notes?

Oh right $\overline{A}$ = NOT (A)

¿So just flip bits?

1 0 0

0 0 1 ✓

0 1 0

c) R^2

-So # of lenght 2 walks

-matlab

~~2~~ 1 1

1 ~~2~~ 1

1 1 ~~2~~

¿ how do manually again?

¿why not the 2s? put as 1s

emailed in to ask

(4)

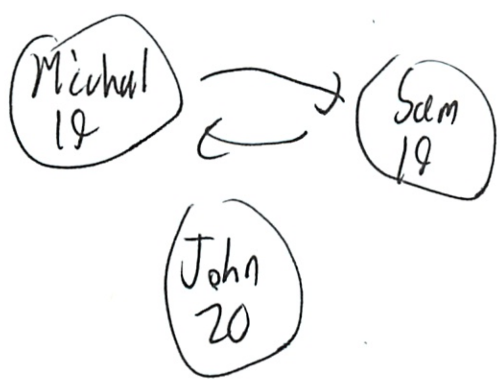
TP.6.3. Relational Properties

1. - reflexive — a R a for all a — path from all vertices to itself
2. - irreflexive — Not (a R a) — no positive lenght path from any vertex to itself (i.no loops)
3. - antisymmetric — a R b → Not (b R a) arrows only go 1 way
4. - transitive — (a R b AND b R c) → a R c

1. are the same age - relation on people

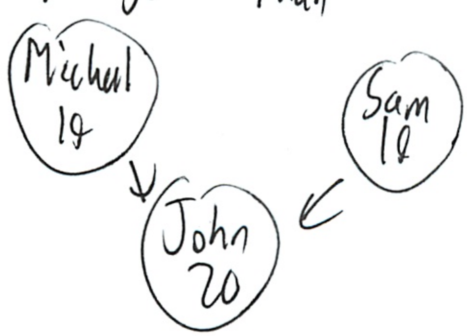
   ↑ So what exactly is this

   a graph from people of same age like



reflexive (can do loops) 1,
Symmetric (don't care)
transitive 4, ✓

2. is younger than

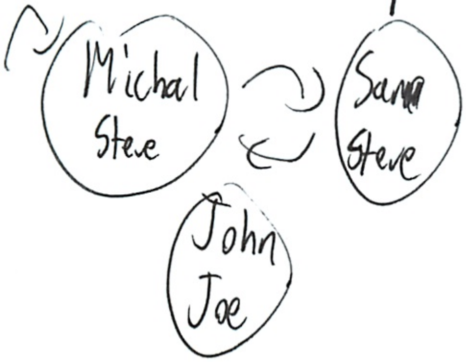


No loops 2
antisem 3
trans 4 ✓

↑ drawing example helpful

(5)

3. have the same parents



1,
Sym ✓
4

4. is decendent of
(should be able to do w/o pic)

1 no
2 ✓ (mentally picture)
3
4

5. have a parent in common
? same as 3 ? ✗

1

not 4 if parents remarry ✓

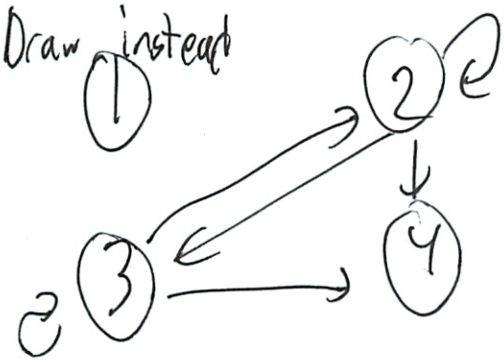
⑥

TP4 Binary Relation properties

Set {1, 2, 3, 4}

r = reflexive
i = irreflexive
a = antisym
t = trans
n = none of above

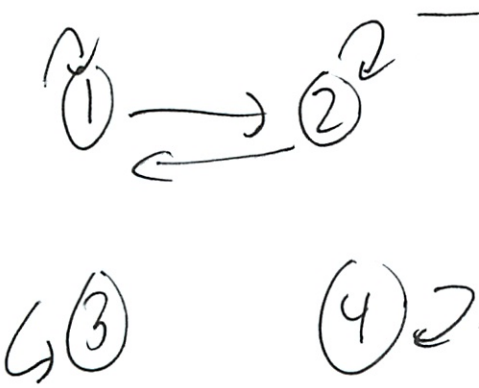
1. Draw instead



r
t i not really directly

r ⊗
r t ⊗
Oh not r, since not all ↻ loops
n ⊗
t ✓

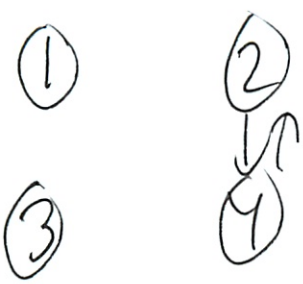
2.



r
t ✓

Is everything transitive?

(7)

3. 

i

not t since not enough arrows?

So there is a not (t) one

4. (p, q) - diff people who speak same lang

∴ i = diff people

always sym

t

it (x)

t (x) <u>not t</u> Some people are bilingual

? how is this - more like not r or i

Just i

Not a: more than 1 person speaks eng

Why not t? all people who speak same lang

? Just 1 lang or share lang set?

5. (A, B) more than 1,000,000 people speak both A and B?

-? is this only 1 lang?

i (x)

h

not r → Some lang spoken by < 1,000,000

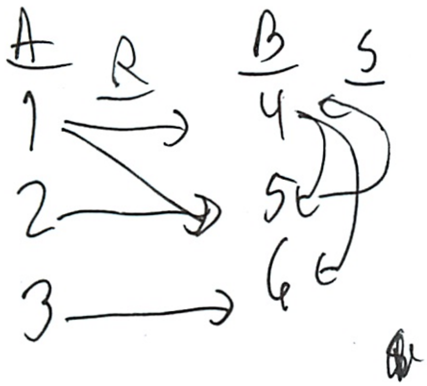
? So set is all lang -oh this set + itself if spoken by over 1,000,000 total people counts

not i → some langs spoken > 1,000,000

not t more than 1,000,000 Czech speakers, eng also and 1,000,000 Chinese + Eng but not 1,000,000 Czech + Chinese

⑧.

TPG.5 Composition of Relations



S is relation B to B

Find S o R

9.4.1 Composition of relations

R : B → C

S : A → B

R o S : A → C ← is not backwards

a (R o S) c ::= ∃ b ∈ B (a S b) and (b R c)

Sets A → B ← not helpful

but directly only if someting in middle

S o R

(a R b) and (b S c)

↑ note direction

1 → 4 → 5 (1 5)

1 → 4 → 6 (1 6)

1 → 5 → 4 (1 4)

2 → 5 → 4 (2 4)

(9)

b) $S \circ S$

$4 \to 5 \to 4$ (4 4)

$5 \to 4 \to 5$ (5 5) ✓

$5 \to 4 \to 6$ (5 6)

c) $S^{-1} \circ R$

S^{-1} is flip arrow (5,4)(6,4)(4,5)

(a R b) AND (b S^{-1} c)

$1 \to 4 \to 5$ (1 5)

$1 \to 5 \to 4$ (1 4) ✓

$2 \to 5 \to 4$ (2 4)

$3 \to 6 \to 4$ (3 4)

6.6 Partial + Total Orders

p = partial but not total

t = total

n = neither

(this was section did not understand)

a. (p,q) p, q are people of same age.

what is whole set?

are they directly comparable? t ⊗ n

(10)

b. (a,b) a is age of someone who is not younger than anyone of age b

¿older

I'm just guessing ... p ⊗
n ⊗
+ What the hell?
Don't get at all

c) (p,q) p is person who is int. mutiple of q's age
p ⊗
+ ⊗
n

---

a) Reflexive and tran but not antisym. So not partial let alone total

b) Ages can translate to days or some numeric unit - so somewhat akward description of ≥ on these #

c) Two diff people can be same age so not asym. ruling out p.t. Note that relation on ages (as opposed to persons) would be same as divisability relation on natural # for which p would be corect ans — a bit of a trick qu

(11)

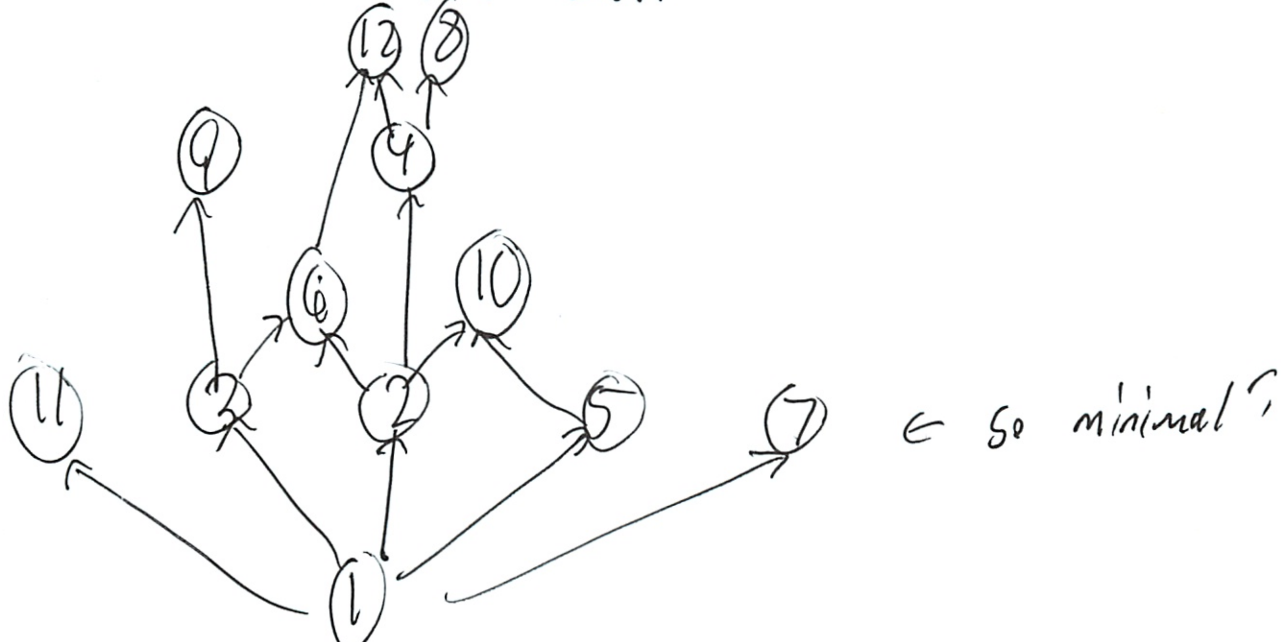
6.7 Partial ordering

The "divides" partial ordering on set

{2, 4, 6, 9, 12, 18, ~~24~~ 27, 36, 48, 60, 72}

2 is smaller than 4, since 2|4

1. What are minimal elements
   - oh things where nothing goes in
   - go back to that chart



But diff # here

2, 9 ✓

2. maximal

72 60 48 ~~36~~ 27 ✓

since $\frac{72}{2} = 36$

(12)

How to do automatically?

- well guess build graph
- and count arrows
  - or look on matrix

3. What are larger than both 2 and 9?

? what does mean - has arrows in from 2,9

(are both parents/granparents)

18 36 72 ✓

Mathematics for Computer Science
MIT 6.042J/18.062J

DAG's & Scheduling

Some Course 6 Prerequisites

18.01 → 6.042
18.01 → 18.02
18.01 → 18.03
6.001 → 6.034
6.042 → 6.046

8.02 → 6.002
18.03, 6.002 → 6.004
6.001, 6.004 → 6.033
6.033 → 6.857
6.046 → 6.840

indirect prerequisites

if sequence of prereq's from u to v, say
u is an "indirect prereq" of v
u is "earlier" than v
"smaller"

indirect prerequisites

so u is an indirect prereq of v just means that there is a positive length path from u to v in the prerequisite digraph R:

$$u\ R^+\ v$$

a minimal subject?

a minimal subject has no preequisites --a Freshman subject

nothing → d

18.01 8.02 6.001

a minimum subject?

minimum means earliest of all: an indirect prereq. of everything
none in this example
there used to be one at MIT: orientation week seminar on on summer book assignment

Constructing a Term Schedule

18.01 → 6.042
18.01 → 18.02
18.01 → 18.03
6.001 → 6.034
6.042 → 6.046

8.02 → 6.002
18.03, 6.002 → 6.004
6.001, 6.004 → 6.033
6.033 → 6.857
6.046 → 6.840

identify minimal elements

Constructing a Term Schedule

18.01 8.02 6.001

start schedule with them

Constructing a Term Schedule

~~18.01~~ → 6.042
~~18.01~~ → 18.02
~~18.01~~ → 18.03
~~6.001~~ → 6.034
6.042 → 6.046

~~8.02~~ → 6.002
18.03, ~~6.002~~ → 6.004
~~6.001~~, 6.004 → 6.033
6.046 → 6.840

remove minimal elements

Constructing a Term Schedule

→ 6.042
→ 18.02
→ 18.03
→ 6.034
6.042 → 6.046

→ 6.002
18.03, 6.002 → 6.004
6.004 → 6.033
6.033 → 6.857
6.046 → 6.840

remove minimal elements

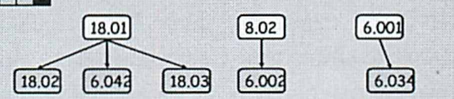
Constructing a Term Schedule

→ 6.042
→ 18.02
→ 18.03
→ 6.034
6.042 → 6.046

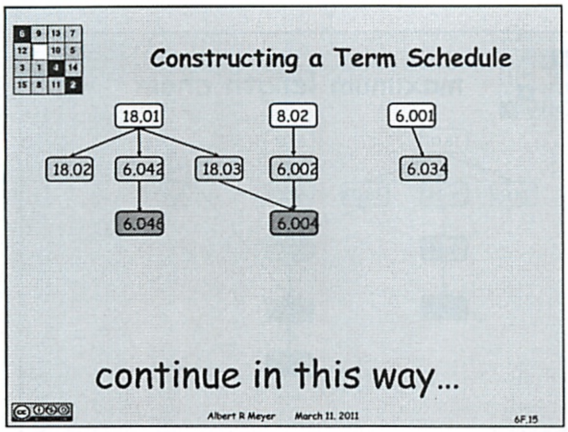
→ 6.002
18.03, 6.002 → 6.004
6.004 → 6.033
6.033 → 6.857
6.046 → 6.840

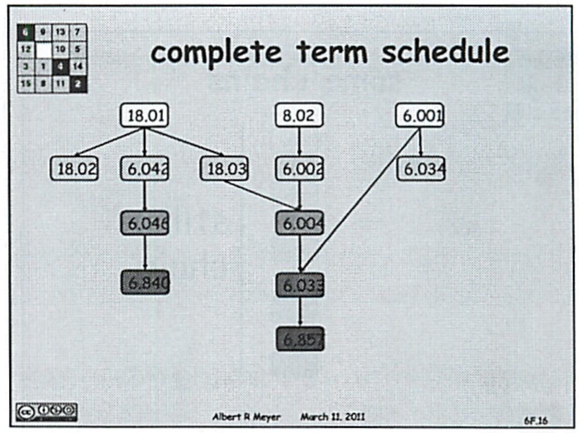
identify new minimal elements

Constructing a Term Schedule



schedule them next





an antichain

a set of subjects with no indirect prereqs among them

--so can be taken in *any order*

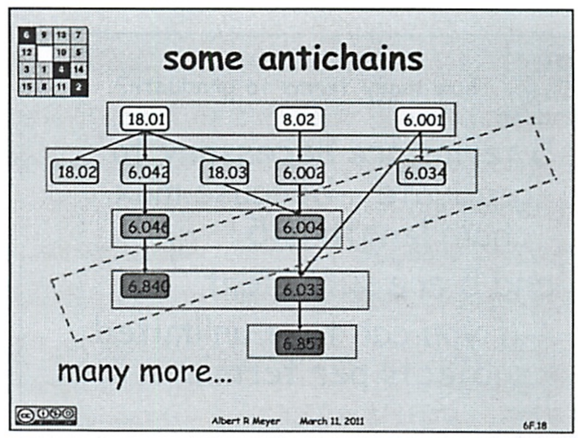
--called "incomparable"

Def: u is incomparable to v iff

no path from u to v and

no path from v to u

Albert R Meyer March 11, 2011 6F.17

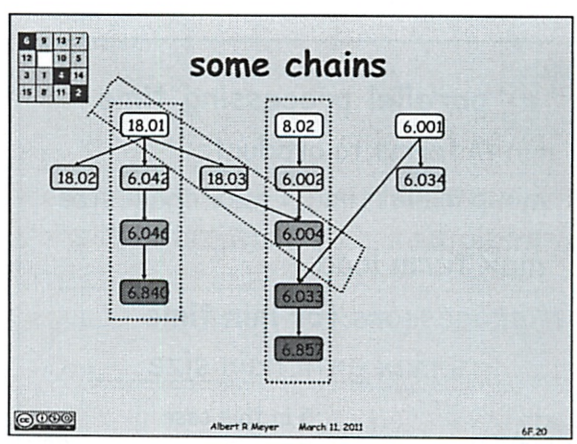


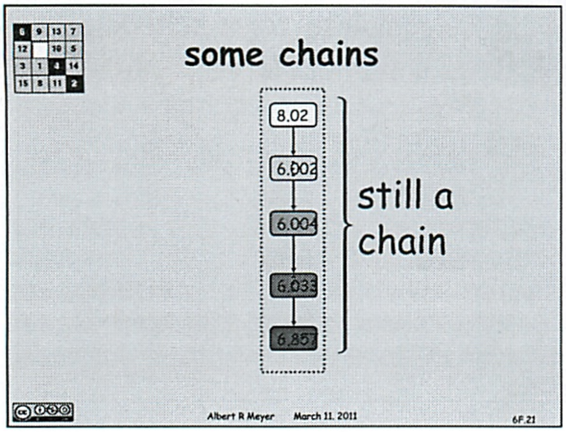
a chain

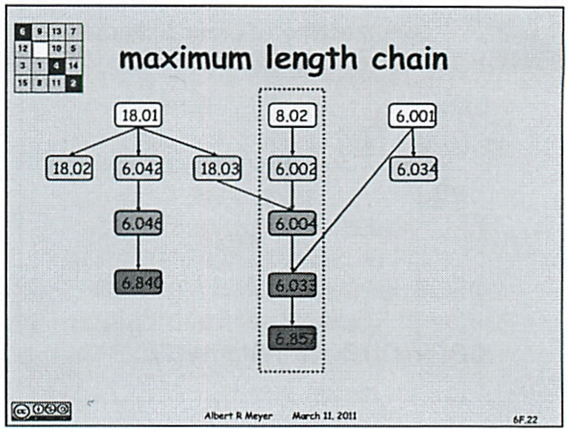
sequence of subjects that must be taken in order

(subjects are *comparable*)

Albert R Meyer March 11, 2011 6F.19







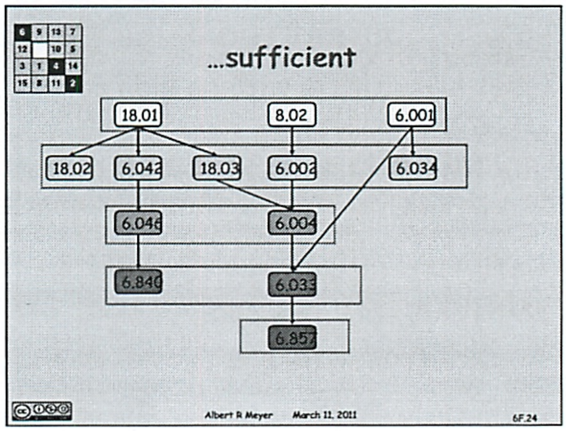
how many terms to graduate?

5 terms are necessary to graduate --because max chain length is 5

and 5 are sufficient

--if you can take unlimited subjects per term...

Albert R Meyer March 11, 2011



parallel processing time

min # terms to graduate:

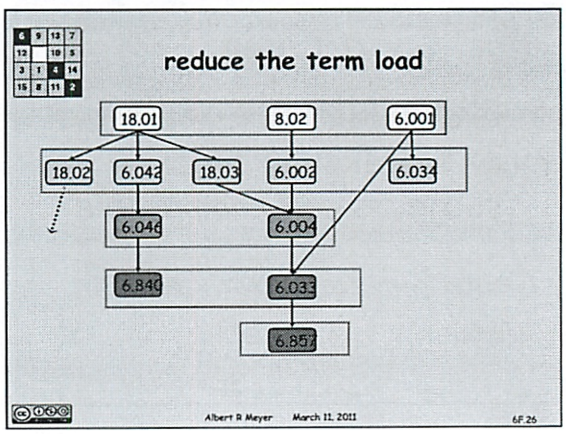
min parallel time = max chain size

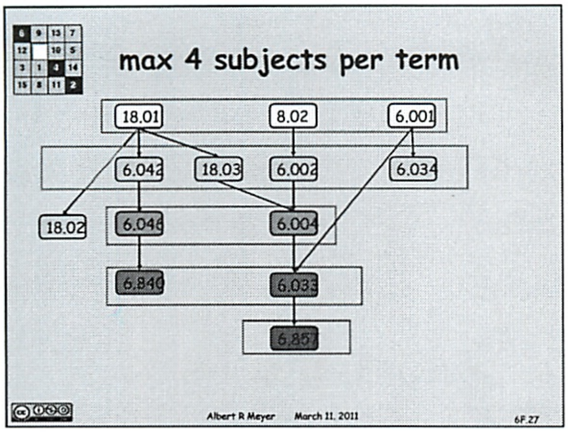
max term load:

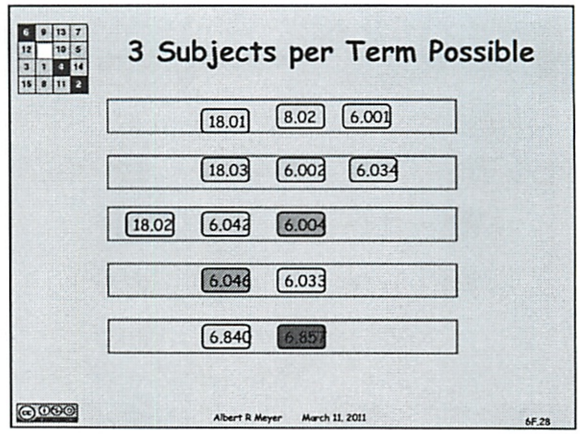
processors for min time ≤ max antichain size

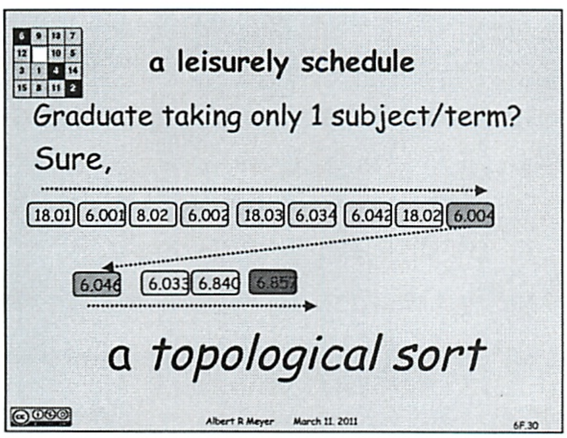
5 in this case

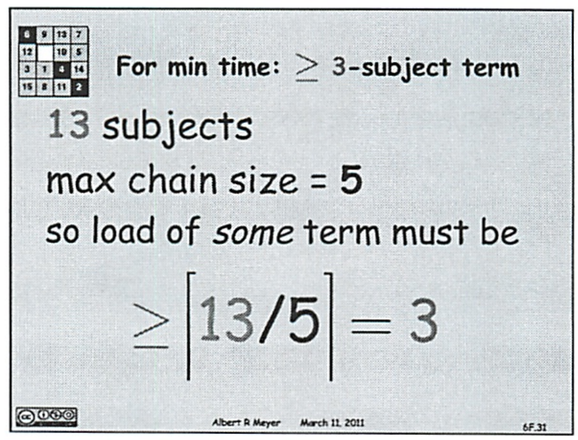
Albert R Meyer March 11, 2011 6F.25

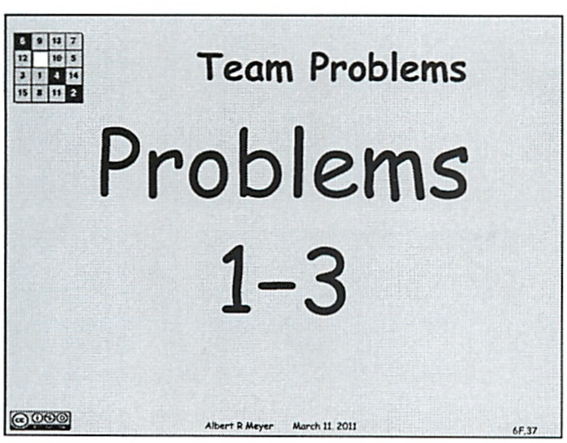



max 4 subjects per term
18.01
8.02
6.001
6.042
18.03
6.002
6.034
18.02
6.046
6.004
6.840
6.033
6.857
Albert R Meyer March 11, 2011
6F.27


3 Subjects per Term Possible
18.01
8.02
6.001
18.03
6.002
6.034
18.02
6.042
6.004
6.046
6.033
6.840
6.857
Albert R Meyer March 11, 2011
6F.28


a leisurely schedule
Graduate taking only 1 subject/term?
Sure,
18.01
6.001
8.02
6.002
18.03
6.034
6.042
18.02
6.004
6.046
6.033
6.840
6.857
a topological sort
Albert R Meyer March 11, 2011
6F.30


For min time: ≥ 3-subject term
13 subjects
max chain size = 5
so load of some term must be
≥ ⌈13/5⌉ = 3
Albert R Meyer March 11, 2011
6F.31


Team Problems
Problems
1–3
Albert R Meyer March 11, 2011
6F.37