CSC165 fall 2019
rooted trees / what’s next

Danny Heap
csc165-2019-09@cs.toronto.edu
BA4270 (behind elevators)
Web page:
http://www.teach.cs.toronto.edu/~heap/165/F19/

Using Course notes: trees
distinguish a root

add notions of distance, hierarchy/direction to trees by

rooted tree: a tree with

- exactly one vertex labelled (distinguished) as root, if the tree has at least one vertex

- OR no vertices (a convenience for proofs and algorithms)
jargon

- parent
- child
- ancestor
- descendant
- arity (branching factor)
- height, denote as $\text{height}(G)$
easy-ish facts

- every rooted tree with $n \geq 2$ vertices has height at least 2
- some rooted tree with $n \geq 2$ vertices has height exactly 2
- every rooted tree with $n$ vertices has height no more than $n$
- some rooted tree with $n$ vertices has height exactly $n$
binary rooted trees
maximum degree 3 \equiv maximum of 2 children
\forall h \in \mathbb{N}, \forall G = (V, E) (G \text{ rooted, binary tree } \land \text{height}(G) \leq h) \Rightarrow |V| \leq 2^h - 1
later topics...

- prove correctness
- analyze recursive runtime
- computability
- intractability
- public-key cryptography
problem with keys... e.g. Vigenere cipher

key: thewalrusandthecarpenter
cleartext: ifsevenmaidswitsevenmopssweptforhalfayear

ifsevenmaidswitsevenmopssweptforhalfayear
thewalrusandthecarpenterthewalrusandthecar

how do you securely exchange keys?
public/private

share public key with the world
keep private key secret

allows:

authentication

encryption
RSA

need: text → integer, integer → text reversible padding scheme

1. randomly choose large primes $p$ and $q$
2. $n = pq$ (key length is $n$ in bits...)
3. $L = (p - 1)(q - 1)$
4. choose $1 < e < L$ so that $\gcd(e, L) = 1$
5. compute inverse, $d \equiv e^{-1} \pmod{L}$, i.e. $de \equiv 1 \pmod{L}$
   (notes Example 2.19 works for co-prime!)

publish: $e, n$

keep private $d, p, q, L$.

$m =$ text → integer(message)

encrypt: $c \equiv m^e \pmod{n}$

decrypt: message = integer → text($c^d$) (mod $n$))
it works... how?
Use results from this course... mostly

- $c^d \equiv m^{ed} \pmod{n}$

- $n = pq$, and $ed \equiv 1 \pmod{(p - 1)(q - 1)}$, i.e. $ed = 1 + k(p - 1)(q - 1)$

- $m^{ed} \equiv m \times m^{(p-1)(q-1)k} \pmod{p} \equiv m \times 1^{(q-1)k} \pmod{p}$
  (problem set #3, Q1(c) almost...) $\equiv m \pmod{p}$

- also $m^{ed} \equiv m \pmod{q}$

- (problem set #2, Q2(a)): $m^{ed} \equiv m \pmod{pq} \equiv m \pmod{n}$. 