CSE 311

Foundations of Computing I

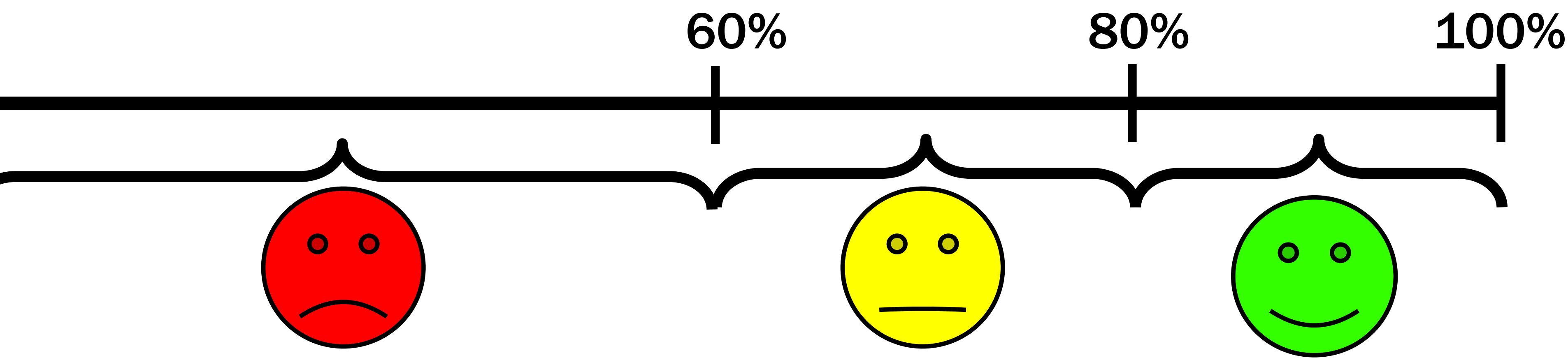
* All slides are a combined effort between previous instructors of the course

HW 3 De-Brief

Think back to when you wrote your first essay.

HW 3 De-Brief

If your HW 3 score is in this range…



Schedule an appointment with Adam today. Not tomorrow; today.

There is something very important you didn't understand. If you aren't 100% on every comment you got by tomorrow, schedule an appointment with Adam.

Great Work! Read all your feedback.

≈18%

≈32%

≈50%

HW 3 De-Brief

Okay, I got it. How do I schedule an appointment?

- Go to http://meeting.countablethoughts.com
- If I don’t respond by Monday, then it probably didn’t go through; so, e-mail me.

HW 3 De-Brief

“How I Oops 311”

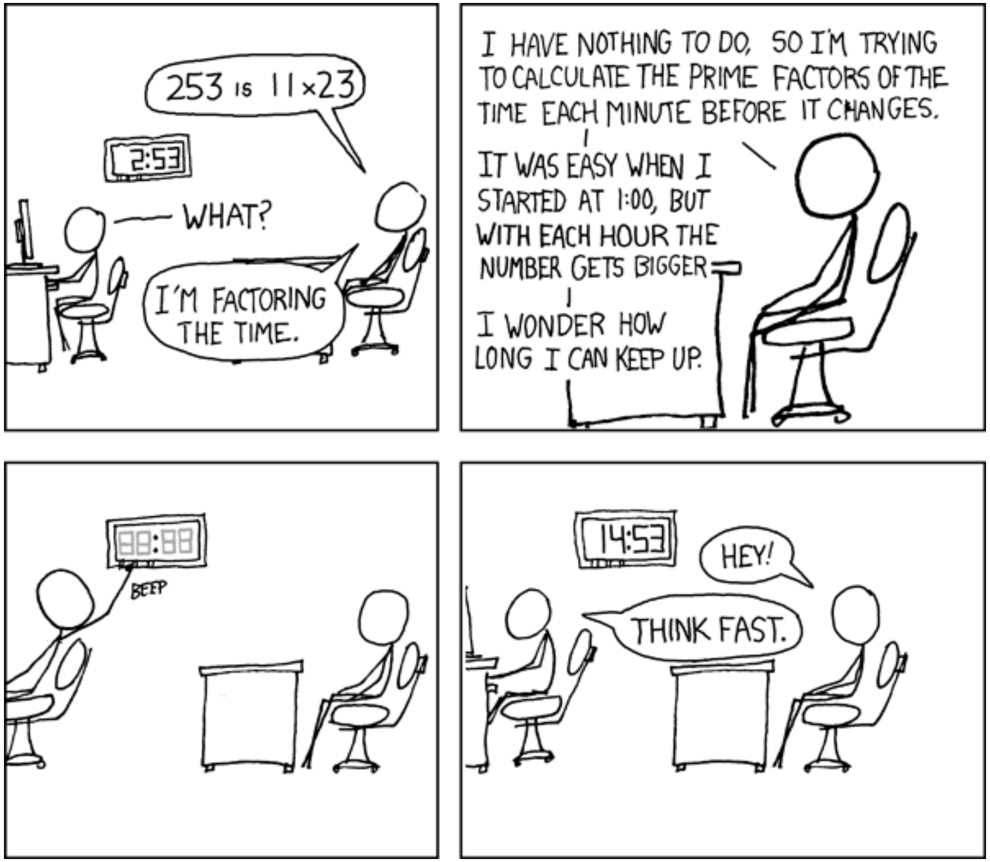
- Never read the feedback, or
- Read the feedback but don’t take it seriously, or
- Read the feedback but convince yourself that “you get it now”, or
- Read the feedback, talk to a TA, but don’t apply what you’ve learned to future HWs, or…

HW 3 De-Brief

How smart you are and your grade are not the same thing.

CSE 311: Foundations of Computing

Lecture 12: Primes, GCD



Sign-Magnitude Integer Representation

n-bit signed integers

Suppose $-2^{n-1} < x < 2^{n-1}$

First bit as the sign, n-1 bits for the value

99 = 64 + 32 + 2 + 1

18 = 16 + 2

For n = 8:

99: 0110 0011

-18: 1001 0010

Any problems with this representation?

Two's Complement Representation

n bit signed integers, first bit will still be the sign bit

Suppose $0 \le x < 2^{n-1}$,

x is represented by the binary representation of x

Suppose $0 \le x \le 2^{n-1}$,

$-x$ is represented by the binary representation of $2^n - x$

Key property: Twos complement representation of any number y is equivalent to y mod 2^n so arithmetic works mod 2^n

99 = 64 + 32 + 2 + 1
18 = 16 + 2

For n = 8:
99: 0110 0011
-18: 1110 1110

Sign-Magnitude vs. Two's Complement

-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7
1111	1110	1101	1100	1011	1010	1001	0000	0001	0010	0011	0100	0101	0110	0111

Sign-bit

-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7
1000	1001	1010	1011	1100	1101	1110	1111	0000	0001	0010	0011	0100	0101	0110	0111

Two's complement

Two's Complement Representation

- For $0 < x \leq 2^{n-1}$, $-x$ is represented by the binary representation of $2^n - x$

- To compute this: Flip the bits of x then add 1:
  - All 1's string is $2^n - 1$, so

    Flip the bits of x ≡ replace x by $2^n - 1 - x$

Basic Applications of mod

- Hashing
- Pseudo random number generation
- Simple cipher

Hashing

Scenario:

Map a small number of data values from a large domain $\{0, 1, \dots, M-1\}$ ...

...into a small set of locations $\{0,1,\dots,n-1\}$ so one can quickly check if some value is present

- $\mathrm{hash}(x) = x \bmod p$ for p a prime close to n
  - or $\mathrm{hash}(x) = (ax+b) \bmod p$
- Depends on all of the bits of the data
  - helps avoid collisions due to similar values
  - need to manage them if they occur

Pseudo-Random Number Generation

Linear Congruential method

$$\boxed{x_{n+1} = (a\, x_n + c) \bmod m}$$

Choose random x_0, a, c, m and produce a long sequence of x_n's

Modular Exponentiation mod 7

x	1	2	3	4	5	6
1	1	2	3	4	5	6
2	2	4	6	1	3	5
3	3	6	2	5	1	4
4	4	1	5	2	6	3
5	5	3	1	6	4	2
6	6	5	4	3	2	1

a	a^1	a^2	a^3	a^4	a^5	a^6
1						
2						
3						
4						
5						
6						

Exponentiation

- **Compute** 78365^{81453}
- **Compute** $78365^{81453} \bmod 104729$
- **Output is small**
  - need to keep intermediate results small

Repeated Squaring – small and fast

Since $a \bmod m \equiv a \pmod{m}$ for any a

we have $a^2 \bmod m = (a \bmod m)^2 \bmod m$

and $a^4 \bmod m = (a^2 \bmod m)^2 \bmod m$

and $a^8 \bmod m = (a^4 \bmod m)^2 \bmod m$

and $a^{16} \bmod m = (a^8 \bmod m)^2 \bmod m$

and $a^{32} \bmod m = (a^{16} \bmod m)^2 \bmod m$

Can compute $a^k \bmod m$ **for** $k=2^i$ **in only** i **steps**

Fast Exponentiation

```
public static long FastModExp(long base, long exponent, long modulus) {
       long result = 1;
       base = base % modulus;

       while (exponent > 0) {
           if ((exponent % 2) == 1) {
               result = (result * base) % modulus;
                exponent -= 1;
           }
           /* Note that exponent is definitely divisible by 2 here. */
           exponent /= 2;
           base = (base * base) % modulus;
           /* The last iteration of the loop will always be exponent = 1 */
           /* so, result will always be correct. */
       }
       return result;
   }
```

$b^e \bmod m = (b^2)^{e/2} \bmod m$, when e is even)

$b^e \bmod m = (b*(b^{e-1} \bmod m) \bmod m)) \bmod m$

Program Trace

Let M = 104729

$78365^{81453} \bmod M$

$= ((78365 \bmod M) * (78365^{81452} \bmod M)) \bmod M$

$= (78365 * ((78365^2 \bmod M)^{81452/2} \bmod M)) \bmod M$

$= (78365 * ((78852)^{40726} \bmod M)) \bmod M$

$= (78365 * ((78852^2 \bmod M)^{20363} \bmod M)) \bmod M$

$= (78365 * (86632^{20363} \bmod M)) \bmod M$

$= (78365 * ((86632 \bmod M) * (86632^{20362} \bmod M)) \bmod M$

$= \ldots$

$= 45235$

Fast Exponentiation Algorithm

Another way:

$$81453 = 2^{16} + 2^{13} + 2^{12} + 2^{11} + 2^{10} + 2^{9} + 2^{5} + 2^{3} + 2^{2} + 2^{0}$$

$$a^{81453} = a^{2^{16}} \cdot a^{2^{13}} \cdot a^{2^{12}} \cdot a^{2^{11}} \cdot a^{2^{10}} \cdot a^{2^{9}} \cdot a^{2^{5}} \cdot a^{2^{3}} \cdot a^{2^{2}} \cdot a^{2^{0}}$$

$$\begin{aligned}
a^{81453} \bmod m &= \\
(\ldots(((((a^{2^{16}} &\bmod m \cdot \\
a^{2^{13}} &\bmod m\ ) \bmod m \cdot \\
a^{2^{12}} &\bmod m) \bmod m \cdot \\
a^{2^{11}} &\bmod m) \bmod m \cdot \\
a^{2^{10}} &\bmod m) \bmod m \cdot \\
a^{2^{9}} &\bmod m) \bmod m \cdot \\
a^{2^{5}} &\bmod m) \bmod m \cdot \\
a^{2^{3}} &\bmod m) \bmod m \cdot \\
a^{2^{2}} &\bmod m) \bmod m \cdot \\
a^{2^{0}} &\bmod m)\ \bmod m
\end{aligned}$$

The fast exponentiation algorithm computes $a^n \bmod m$ using $O(\log n)$ multiplications $\bmod m$

Primality

An integer *p* greater than 1 is called *prime* if the only positive factors of *p* are 1 and *p*.

A positive integer that is greater than 1 and is not prime is called *composite*.

Fundamental Theorem of Arithmetic

Every positive integer greater than 1 has a unique prime factorization

48 = 2 • 2 • 2 • 2 • 3
591 = 3 • 197
45,523 = 45,523
321,950 = 2 • 5 • 5 • 47 • 137
1,234,567,890 = 2 • 3 • 3 • 5 • 3,607 • 3,803

Euclid's Theorem

There are an infinite number of primes.

Proof by contradiction:

Suppose for contradiction that there are n primes for some natural number n. Call them $p_1 < p_2 < \ldots < p_n$. Consider $P = p_1 p_2 \ldots p_n$, and define $Q = P + 1$.

Case 1 (Q is prime). Then, we're done, because Q is larger than any of the primes; so, it is a new prime.

Case 2 (Q is composite). Then, there must be some prime p such that $p \mid Q$. Note that since P divides every possible prime, $p \mid P$ as well. It follows that $p \mid (Q - P) \rightarrow p \mid ((P + 1) - P) \rightarrow p \mid 1$. This is impossible, because p must be at least two.

Since both cases lead to a contradiction, the original claim is true.

Famous Algorithmic Problems

- Primality Testing
  - Given an integer n, determine if n is prime
- Factoring
  - Given an integer n, determine the prime factorization of n

Factoring

Factor the following 232 digit number [RSA768]:

1230186684530117755130494958384962720772
8535695953347921973224521517264005072636
5751874520219978646938995647494277406384
5925192557326303453731548268507917026122
1429134616704292143116022212404792747377
94080665351419597459856902143413

12301866845301177551304949583849627207728535695953347
92197322452151726400507263657518745202199786469389956
47494277406384592519255732630345373154826850791702612
21429134616704292143116022212404792747377940806653514
19597459856902143413

=

33478071698956898786044169848212690817704794983713768568912431388982883793878002287614711652531743087737814467999489

×

36746043666799590428244633799627952632279158164343087642676032283815739666511279233373417143396810270092798736308917

Factoring

Uh…fun?

Greatest Common Divisor

GCD(a, b):

Largest integer d **such that** $d \mid a$ **and** $d \mid b$

- GCD(100, 125) =
- GCD(17, 49) =
- GCD(11, 66) =
- GCD(13, 0) =
- GCD(180, 252) =

GCD and Factoring

$a = 2^3 \cdot 3 \cdot 5^2 \cdot 7 \cdot 11 = 46{,}200$

$b = 2 \cdot 3^2 \cdot 5^3 \cdot 7 \cdot 13 = 204{,}750$

$$GCD(a, b) = 2^{\min(3,1)} \cdot 3^{\min(1,2)} \cdot 5^{\min(2,3)} \cdot 7^{\min(1,1)} \cdot 11^{\min(1,0)} \cdot 13^{\min(0,1)}$$

Factoring is expensive!

Can we compute GCD(a,b) without factoring?