CSC104 winter 2013

Why and how of computing

week 1

Danny Heap

heap@cs.toronto.edu ← good contact method

BA4270 (behind elevators) ← better

http://www.cdf.toronto.edu/~heap/104/W13/ ← URL for course

416-978-5899 ← occasionally answer...

Text: Picturing Programs

Outline

Who needs to know why and how?

computer ≠

- computer is general-purpose, toaster makes bread warm

- you (usually) shouldn't tinker with a toaster, but you should with a computer

- We all consume computing, the thing is to change it
- Computers and networks change society — privacy, property, democracy, work, education — for better or worse
- We get an insight into computer culture by making some artifacts: programs

Two tracks in this course

- Insight into computing mindset: problem-solving and programs → "gentle introduction to programming"

- History of computing technology, overview of modern computing OS, social issues

How to do well at this course

- Read the course information sheet as a two-way promise

  becomes final 11/1/13

- humour me: read your email

  I send out announcements

- Question, answer, record, synthesize

  write on your copy of these slides.

- Collaborate with respect

What to do with computing machines?

Algorithms!

simple enough for a machine to follow

simple sequence of feasible steps to solve a problem

deterministic (in this course)

same instructions → same result

credit Al-Khwarizmi

Examples

- multiplication
- PBJ
- Google page rank

Sticky algorithm

pbj

peanut butter bread jam → PBJ sandwich
could you explain it to a friend
over the phone, who had
never made it?

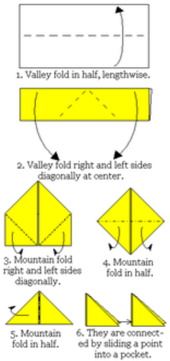
Fraught with peril!

- which operations are built-in? ← "open jar"
- what if conditions change? → PB had foil liner!
- name repeated operations ← "scoop from jar"
- does sequence matter? → scoop before jar is open?

paper folding

(crease-pattern 5)

→ 



(ignore the diagram on the left)
fold over upper surface of paper strip
after one fold, it has a downward crease
fold the once-folded strip again
and it has one upward, two downward
there are good physical reasons you
can't experiment far beyond 6 folds
given the number of folds,
predict the pattern

For more information, and hints, see paper folding problem

2000+ year-old algorithm

(GCD 6 15)

Euclid's GCD

the largest whole number that divides two non-negative whole numbers is their Greatest Common Denominator (GCD)

we could find it by sifting through all the divisors, but there's a quicker way

remainder of $n1 \div n2$

e.g (remainder 7 4)

3

(4 goes into 7 once with remainder 3)

Euclid noticed that (gcd $n1$ $n2$) = (gcd $n2$ (remainder $n1$ $n2$))

Also, (gcd $n1$ 0) = $n1$. Repeat as needed.

Try (gcd 15 35)

The way we were

grade school multiplication

4
36
× 47

252
...

×	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7	8	9
2	0	2	4	6	8	10	12	14	16	18
3	0	3	6	9	12	15	18	21	24	27
4	0	4	8	12	16	20	24	28	32	36
5	0	5	10	15	20	25	30	35	40	45
6	0	6	12	18	24	30	36	42	48	54
7	0	7	14	21	28	35	42	49	56	63
8	0	8	16	24	32	40	48	56	64	72
9	0	9	18	27	36	45	54	63	72	81

We'd memorize, and organize, the algorithm for 27×38

Much better than XXVII $\times$ XXXVIII

Notes