Specifically discrete problems, modulo and gcd

Olli-Pekka Hämäläinen

Discrete problem types

- In discrete mathematics, problems can be divided to three main categories depending on their type
  - Existence
  - Counting
  - Optimization
- Each of these problems require different tools
- Note that the same task (in principle) can be handled as any of these (depending on our level of interest)
- Finding a solution could spark change to another type
  - Existence problem → solved: solutions exist

    → “how many?” (counting)

    → “what’s the best one”? (optimization)

Existence problems

- Form: “is there such a solution x that fulfills condition(s) C_i?”
- Conditions can take different forms
  - Equations or inequalities
- Finding a solution is hard
- Checking if a solution is valid is easy
- Number of possible solutions: 0…infinite
- A single one-variable equation is basically a simple existence problem:

$$C_1: x^2 + 10 = 7x \qquad \rightarrow x = 2 \; or \; x = 5$$

- Simple-looking existence problems can be difficult:
  - Sum of three cubes:

$$C_1: x^3 + y^3 + z^3 = k; \quad x, y, z, k \in \mathbb{Z}$$

Counting problems

- Form: “in how many ways can x be done?”
- Often interesting when trying to evaluate the number of possible combinations/operations needed (and hence, estimate the expected solving time)
  - Feasibility of brute-force approach
  - Complexity of an algorithm
- Game theory, probability calculations, etc.
- Break the process in steps, then find out how many ways there are to execute each step
  - Significance of order (does order of steps matter?)
- Usually involves combinations and permutations
- Example: Ford Focus comes in four engine options, two gearbox options, three trim levels and six color options. How many different cars can a customer order?

$$4 \cdot 2 \cdot 3 \cdot 6 = 144$$

Optimization problems

- Form: “which solution x that fulfills condition(s) C_i produces the best result in terms of objective function(s) O_j?”
- Minimization or maximization
- Hard to solve, hard to check
- Discrete optimization relies mostly in identifying the most promising ones among *N* solution candidates
  - *N* can be finite or infinite
- Engineering approach:
  - Treat variables as continuous → round to discrete values OR
  - Decrease *N* using heuristics → brute-force solution
  - Absolute optimum is not necessarily needed
- Mathematical approach:
  - Specific algorithms*
  - Absolute optimum

*Discrete optimization is considered more on course “BM20A8200 Optimointi”.

Example: Linear optimization

- A factory is producing cheap furniture from scrap wood of a nearby sawmill. The factory relies in cheap labor and narrow but perfected product range that consists of only tables and chairs. Material and labor resources for each week as well as what each product requires are presented in the table below.

	Table	Chair	Available
Wood (kg)	31	18	2700
Workhours	6	10	1185

- How many tables and how many chairs should the factory produce in order to optimize profits, if the net profit of tables and chairs are $30 and $40, respectively?

Example: Linear optimization

- Number of tables & chairs must be integers, so this is discrete optimization
- Let's use engineering approach and treat the variables (x = tables, y = chairs) as continuous
- This allows us to write the problem as a standard linear problem, where our objective function is

$$MAX\ O(x, y) = 30x + 40y$$

- Constraint equations from wood & workhour resources:

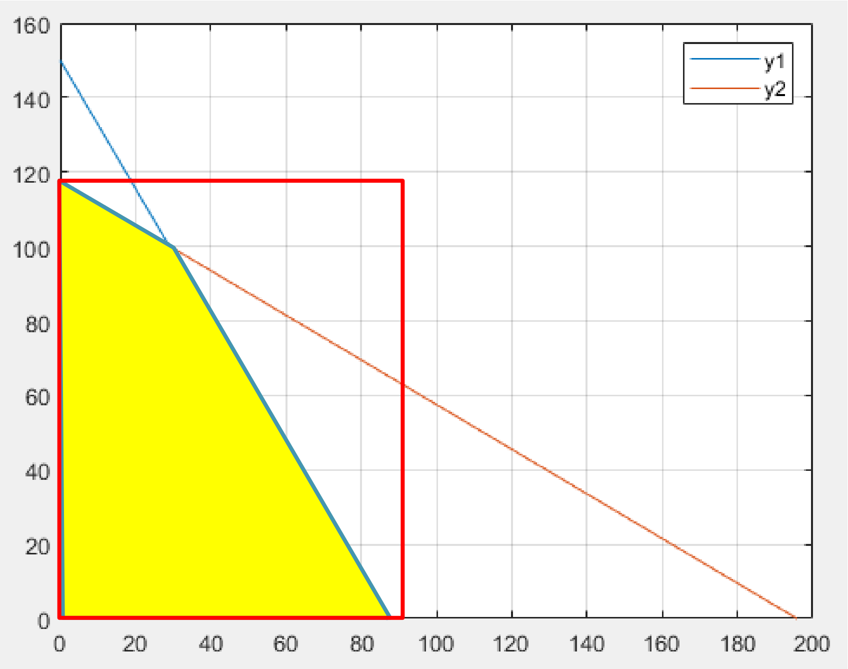
$$\begin{cases} 31x + 18y \leq 2700 \\ 6x + 10y \leq 1185 \end{cases}$$

- Solve y from constraint equations:

$$\begin{cases} y_1 \leq -\dfrac{31}{18}x + 150 \\ y_2 \leq -\dfrac{3}{5}x + 118.5 \end{cases}$$

Example: Linear optimization

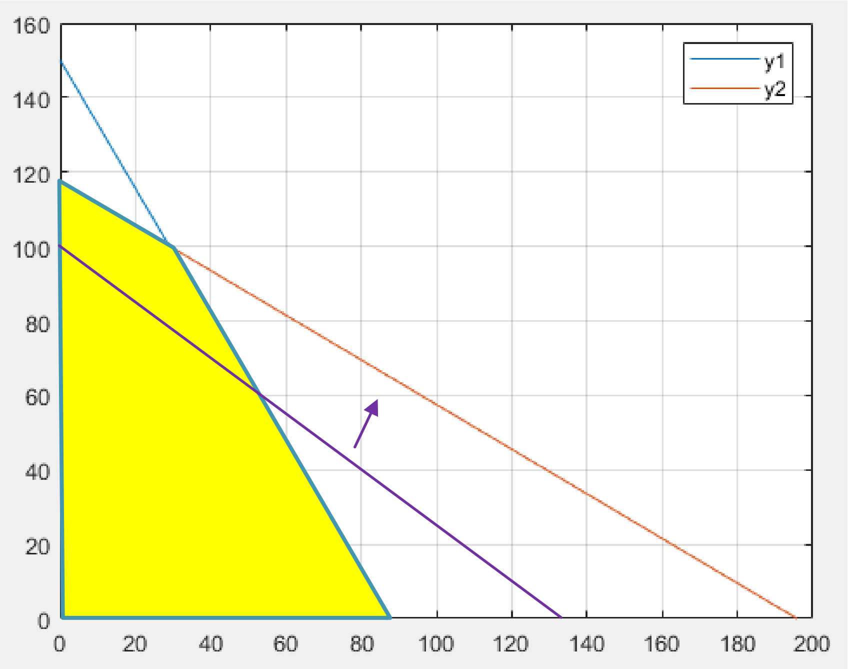
- Draw constrains and define the feasible solution area:



- Judging by the picture, $x \in [0,90]$ and $y \in [0,120]$
  - Brute-force method: 91 x 121 = 11 011 (x,y)-combinations
  - Check feasibility of each combination → calculate O(x,y)

Example: Linear optimization

- We can do better! Let’s use the objective function:



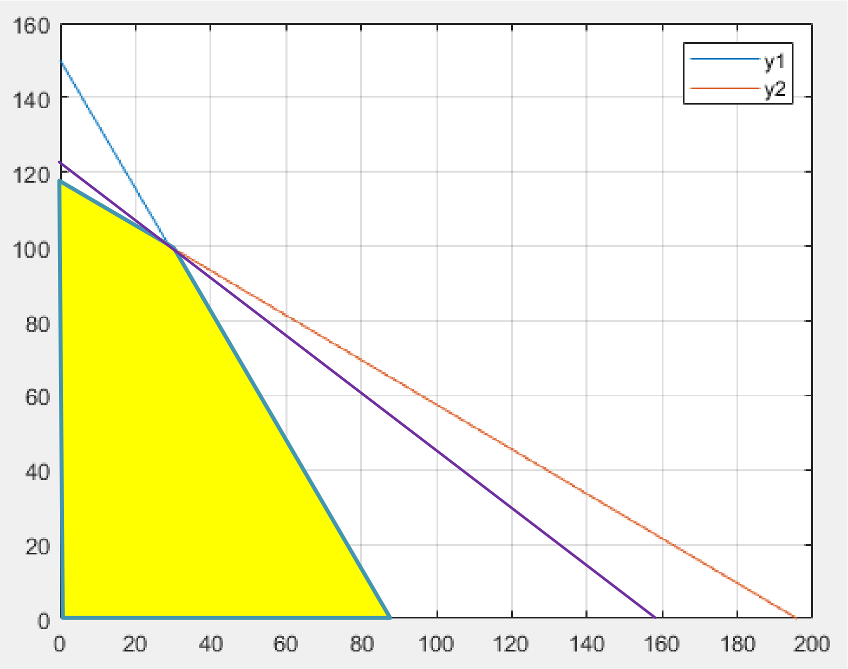
(Here C=4000)

- Draw the contour of objective function and move it upwards until we find the last point where it touches the feasible solution area:

$$30x + 40y = C \;\rightarrow\; y_T = -\frac{30}{40}x + \frac{C}{40}$$

Example: Linear optimization

- We can do better! Let’s use the objective function:



Optimum seems to be at the intersection of y_1 and y_2 (Here C=?)

- Draw the contour of objective function and move it upwards until we find the last point where it touches the feasible solution area:

$$30x + 40y = C \quad \rightarrow \quad y_T = -\frac{30}{40}x + \frac{C}{40}$$

Example: Linear optimization

- Solve the coordinates of the intersection:

$$y_1 = y_2$$

$$-\frac{31}{18}x + 150 = -\frac{3}{5}x + 118.5 \qquad \rightarrow \quad x \approx 28.07$$

$$y = -\frac{3}{5} \cdot 28.07 + 118.5 \approx 101.66$$

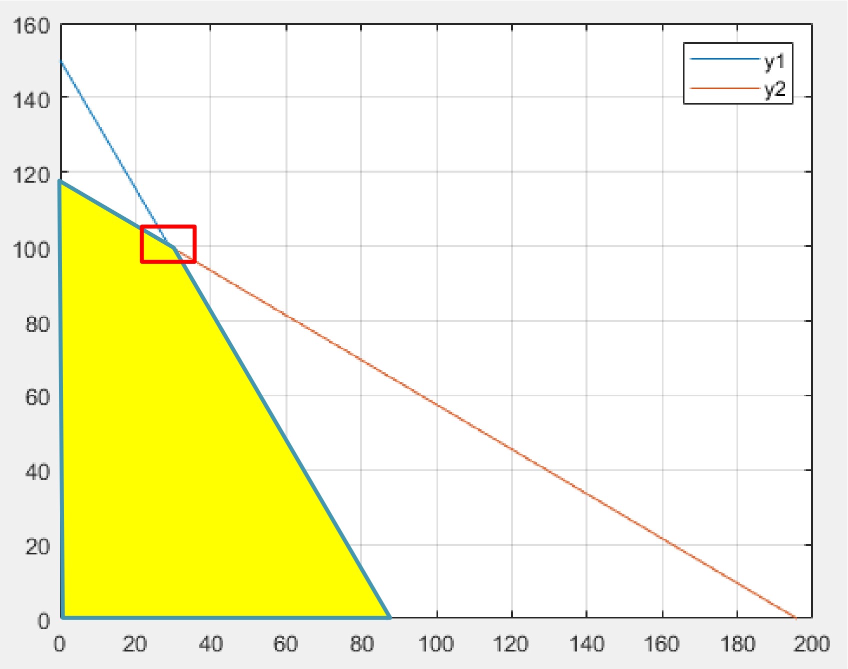
- It would be tempting to round the answer to (28,102). Anyhow, if we do this and substitute to constraints…

$$\begin{cases} 31 \cdot 28 + 18 \cdot 102 = 2704 & > 2700 \\ 6 \cdot 28 + 10 \cdot 102 = 1188 & > 1185 \end{cases}$$

- …we see that this solution violates the constraints.
- What is the optimal discrete solution, then?

Example: Linear optimization

- Let’s go back to our picture and bound tighter:

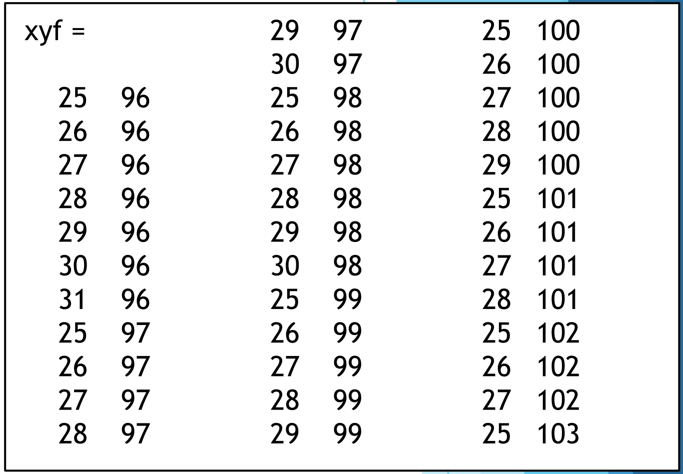


- Judging by the picture, $x \in [25,32]$ and $y \in [96,105]$
  - Brute-force method: 8 x 10 = 80 (x,y)-combinations
  - Check feasibility of each combination → calculate O(x,y)

Example: Linear optimization

- Feasibility check leaves us the following combinations:
- Calculate O(x,y) for all of these

  → find the maximum
- As a result, we get that $O_{max} = 4890$
  - This will happen when x = 27 and y = 102
- Conclusion: **27 tables and 102 chairs!**

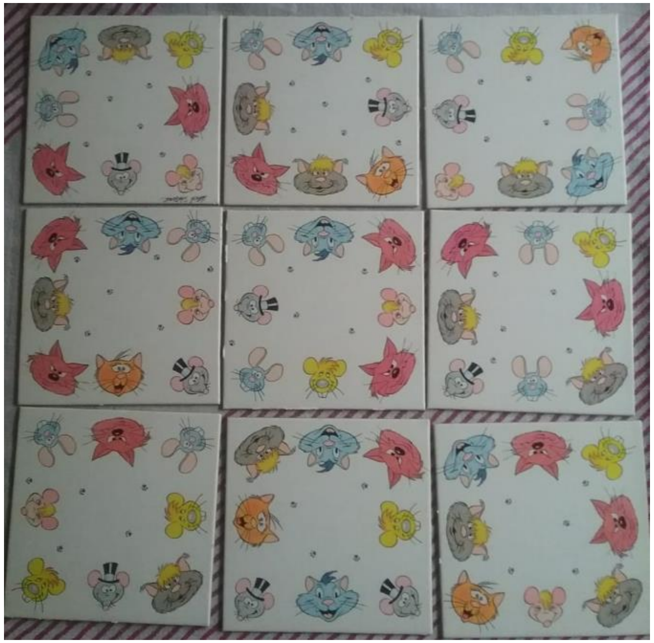


```
xyf =                    29    97        25   100
                         30    97        26   100
   25    96              25    98        27   100
   26    96              26    98        28   100
   27    96              27    98        29   100
   28    96              28    98        25   101
   29    96              29    98        26   101
   30    96              30    98        27   101
   31    96              25    99        28   101
   25    97              26    99        25   102
   26    97              27    99        26   102
   27    97              28    99        27   102
   28    97              29    99        25   103
```

- Takeaways:
  - Continuous analysis enables us more tools (equation solving, function analysis, derivation/integration etc)
  - Problems arise when solutions are not integers
  - Continuous solution MAY help a lot with bounding our solution candidate set and hence deacrease the *N*

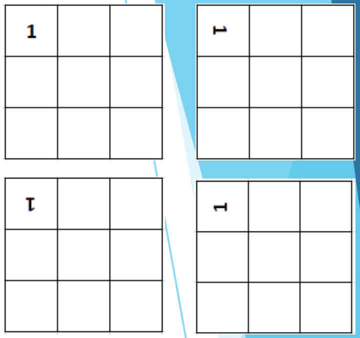
Example: “Cat & mouse” puzzle

- Idea: 9 squares, each has 8 figures of either a cat or a mouse (1 on each side and 1 in each corner)
- Objective: construct a 3x3 square in such a way that cat and mouse are never face-to-face
- Counting problem: how many different 3x3 squares are there? (Only one of them is correct.)



Example: “Cat & mouse” puzzle

- Let’s break the assembly to steps:
  - Step 1: choose top left corner piece (9 options)
  - Step 2: choose orientation of the piece (4 options)
  - Step 3: choose top middle piece (8 options)
  - Step 4: choose orientation of the piece (4 options)
  - Step 5: choose top right corner piece (7 options) etc.



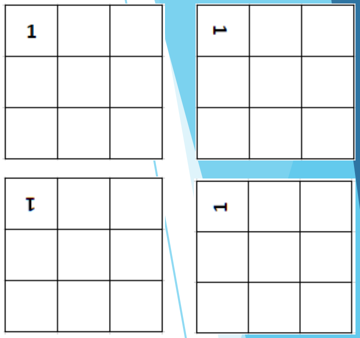
- Total 18 steps needed until all pieces are in place
- Total number of possibilities:

$$9 \cdot 4 \cdot 8 \cdot 4 \cdot 7 \cdot 4 \cdot 6 \cdot \cdots \cdot 1 \cdot 4 = 9! \cdot 4^9 = 362\,880 \cdot 262\,144$$

$$\approx 9.5127 \cdot 10^{10}$$

Example: “Cat & mouse” puzzle

- Let’s break the assembly to steps:
  - Step 1: choose top left corner piece (9 options)
  - Step 2: choose orientation of the piece (4 options)
  - Step 3: choose top middle piece (8 options)
  - Step 4: choose orientation of the piece (4 options)
  - Step 5: choose top right corner piece (7 options) etc.



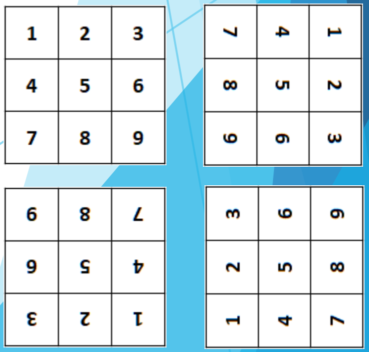
- Total 18 steps needed until all pieces are in place
- Total number of possibilities:

$$9\cdot 4\cdot 8\cdot 4\cdot 7\cdot 4\cdot 6\cdot \dots \cdot 1\cdot 4 = 9!\cdot 4^9 = 362\,880\cdot 262\,144$$

$$\approx 9.5127\cdot 10^{10}$$

- ...but wait a minute: we forgot the global orientation of the 3x3 square! (4 equivalent options)

$$\to \boldsymbol{N} = \frac{\mathbf{9!\cdot 4^9}}{\mathbf{4}} \approx \mathbf{23.78}\ \boldsymbol{billion\ combinations}$$



Example: Block puzzle

- Idea: 3x4 kids puzzle (3yo+), different picture on each side of the cube (= 6 jigsaw puzzles in one)
- Objective: build airplane jigsaw in correct orientation
- Counting problem: in how many ways can we construct the puzzle pieces?

Example: Block puzzle

- Now this is a 3-dimensional puzzle. Break into steps:
  - Step 1: choose top left corner block (12 options)
  - Step 2: choose face of the block (6 options)
  - Step 3: choose orientation of the block (4 options)
  - Step 4: choose top 2nd from left block (11 options)
  - Step 5: choose face of the block (6 options)
  - Step 6: choose orientation of the block (4 options) etc.
- Total number of possibilities:

$$12 \cdot 6 \cdot 4 \cdot 11 \cdot 6 \cdot 4 \cdot \cdots \cdot 1 \cdot 6 \cdot 4 = 12! \cdot 6^{12} \cdot 4^{12} \approx \mathbf{1.7493 \cdot 10^{25}}$$

Takeaways from puzzle examples

- It doesn’t require a large puzzle to make *N* so large that brute-force solution becomes infeasible
- For example, solving the block puzzle would require months of calculation time from a computer
  - …but a 3-year-old kid can do it in less than 10 minutes
- This demonstrates well the power of human brain and its capabilities in heuristics!
- Efficient solution by a machine requires computer vision & pattern recognition in order to decrease the *N*
  - Go through all faces of blocks (12x6 = 72 faces)
  - Identify the candidates that might resemble an airplane and identify the assumed orientation
- Cat & mouse puzzle on the other hand is hard for humans too, because we can’t see if our selections are correct until the last pieces start falling to place

Divisibility problems

- One common category of discrete mathematics problems is questions of divisibility – that is, is x divisible by y, or if it isn't, what's the remainder?
- Division algorithm: every integer x can be expressed as

$$x = qy + r \qquad 0 \leq r \leq |y|$$

  - q = quotient, r = remainder (note! Always nonnegative!)
  - If x is divisible by y, r = 0
- Mechanical calculation of q and r:
  - $q = floor({}^{x}/_{y})$ (floor is a function that rounds the result downwards)
  - $r = x - qy$
- Example: x = 7352, y = 87
  - $q = \frac{7352}{87} \approx 84.506 \quad \rightarrow q = 84$
  - $r = 7352 - 84 \cdot 87 = 44$

Modulo operator

- Quotients and especially remainders are needed fairly often, so in order to ease calculation, we use a “special tool” called modulo operator
  - Computes the remainder: $r = x \, mod \, y$ or $r = mod(x, y)$
  - Very basic command in programming languages
- This operation can be found in most smart calculators
- Also several modulo calculators in internet
- Example: 123812 mod 3128 = 1820
- We’ll use this operator from now on
  - If you don’t have it in your calculator, no panic – you can always calculate the remainder manually (see previous slide) - or use Matlab etc.

Congruence

- In mathematics, congruence means that some items are of same size and shape
- In number theory it is said that two numbers (say a and b) are congruent to modulo m if their remainder is the same when divided by m
- Mathematically speaking, congruence condition is

$$a \bmod m = b \bmod m \quad OR \quad (a - b) \bmod m = 0$$

- If these (equivalent) conditions are met, then a and b are congruent to modulo m; notation for this is

$$a \equiv b \pmod{m}$$

- Often arises with unit system conversions
  - Time (modulo 12 if AM/PM, or modulo 24)
  - Weekdays (modulo 7)
- Also common in cryptography & error detection

Example: ISBN

- All books published in the world since 1972 have carried an ISBN (International Standard Book Number)
- This code was 10 digits long – or actually 9; the last digit was a check digit *s* (0…9 or X, representing 10)
- Calculation formula for the check digit was simple:
  - 1st digit has a weight of 10, 2nd digit a weight of 9 and so on – hence, the weight of the check digit (10th) was 1
  - Sum of all digits d_i multiplied by their weight had to be divisible by 11
  - Mathematically speaking:

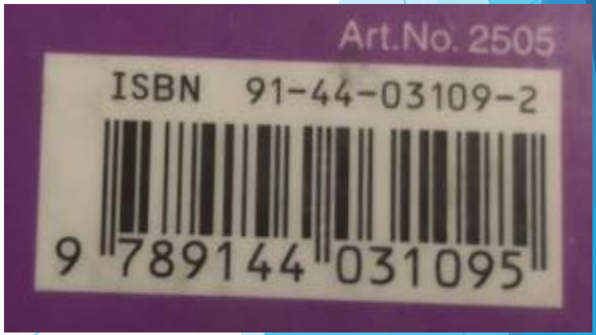
$$\underbrace{\sum_{i=1}^{9}(11-i)d_i}_{=S_d} + s \equiv 0\ (mod\ 11)$$

$$\Rightarrow \quad \boldsymbol{s = 11 - S_d\ mod\ 11}$$

Example: ISBN

- What’s the point?
- When ISBN numbers are written by hand or typed to computer, mistakes may occur
- If a wrong ISBN is typed to a system in a library or a store, it produces an error message (instead of the book being registered incorrectly)
- It can be shown that each valid ISBN differs from others by at least 2 digits, so making a typo in 1 digit will be noticed (most likely others, too)

Note: this original ISBN-10 system was extended to ISBN-13 in 2007. The principle is the same, but the weights of digits are not as logical and ISBN-13 is modulo 10 (so, the check digit can’t be X anymore – just 0…9).
Many books published near the system change date carry both numbers.



Greatest common divisor (gcd)

- When we’re simplifying integer expressions, one commonly performed task is to find the greatest common divisor of two integers a and b
  - In plain English: largest possible value of d so that divisions a/d and b/d will still be integers
- Mathematically this d is marked as gcd(a,b) - or gcd(b,a), since the order is not important here*
- Like modulo operator, this is also found in most programming languages (including Matlab)
  - Smart calculators and internet calculators, too
- Example: gcd(416,338) = 26
- …but how is it actually calculated?

*By definition, it isn’t, but many calculators and algorithms want the input in form a > b.

Euclidean algorithm

- If we don’t have suitable calculators nearby, the greatest common divisor of two nonnegative* integers can be found by using the Euclidean algorithm
- The algorithm is simple and quick to perform:
  1. Initialization: draw table with three columns (a,b,r)
  2. Set the integer values a & b on first row so that a > b
  3. Calculate r = a mod b
  4. On the next row, set $a_{i+1} = b_i$ and $b_{i+1} = r_i$
  5. Calculate r again; repeat steps 4 & 5 until $r_k = 0$
  6. Now we know that $\gcd(a, b) = b_k$; stop

*If our integer(s) are negative, just omit the minus. Divisors will be the same anyway.

Example: gcd(518,182)

- Initialize the table:

a	b	r
518	182	

- Calculate r-values as we proceed:

$r_1 = 518 \, mod \, 182 = 154$

Example: gcd(518,182)

- Initialize the table:

a	b	r
518	182	154
182	154	

- Calculate r-values as we proceed:

$r_1 = 518\ mod\ 182 = 154$

$r_2 = 182\ mod\ 154 = 28$

Example: gcd(518,182)

- Initialize the table:

a	b	r
518	182	154
182	154	28
154	28	

- Calculate r-values as we proceed:

$r_1 = 518\ mod\ 182 = 154$

$r_2 = 182\ mod\ 154 = 28$

$r_3 = 154\ mod\ 28 = 14$

Example: gcd(518,182)

- Initialize the table:

a	b	r
518	182	154
182	154	28
154	28	14
28	14	

- Calculate r-values as we proceed:

$r_1 = 518 \, mod \, 182 = 154$

$r_2 = 182 \, mod \, 154 = 28$

$r_3 = 154 \, mod \, 28 = 14$

$r_4 = 28 \, mod \, 14 = 0$

Example: gcd(518,182)

- Initialize the table:

a	b	r
518	182	154
182	154	28
154	28	14
28	$\boxed{14}$	0

- Calculate r-values as we proceed:

$r_1 = 518 \; mod \; 182 = 154$

$r_2 = 182 \; mod \; 154 = 28$

$r_3 = 154 \; mod \; 28 = 14$

$r_4 = 28 \; mod \; 14 = 0$

➡ $\mathbf{gcd(518, 182) = 14}$

Linear equation ax + by = c

- The gcd can be very helpful for us when solving existence problems
- Problem: is there an integer solution $(x, y \in \mathbb{Z})$ for a linear equation $ax + by = c$ where factors $(a, b, c \in \mathbb{Z})$?
- Answer: a solution exists if and only if $c\ mod\ (\gcd(a, b)) = 0$
  - Justification: if a and b are divisible by d, the terms ax/d and by/d will be integers. Their sum must then be an integer too, so c/d must be an integer. If it isn’t, there can be no solution.
- Example: is there an integer solution for the equation $736x - 464y = 15344$?
  - gcd(736,464) = 16
  - 15344 mod 16 = 0
  - Conclusion: yes, there is!
- …but how could we find that solution? (Or at least one of them, since there may be multiple)

Extended Euclidean algorithm

- We can solve the previous problem by making modifications to our previous Euclidian algorithm
- The modified form produces us integers x_k and y_k so that $ax_k + by_k = \gcd(a, b)$
- The solutions to the original equation can then be found by multiplying these x_k- and y_k-values by $\frac{c}{\gcd(a,b)}$
- Algorithm needs a table with 5 columns: i, q_i, r_i, x_i, y_i
- Initializing the table:
  - Row numbers (i) start from -1
  - First row: $r_{-1} = a, \quad x_{-1} = 1, \quad y_{-1} = 0$
  - Second row: $r_0 = b, \quad x_0 = 0, \quad y_0 = 1$
- Next row values are calculated in recursive fashion:

$q_i = floor\left(\frac{r_{i-2}}{r_{i-1}}\right)$ $r_i = r_{i-2} - q_i r_{i-1}$ $x_i = x_{i-2} - q_i x_{i-1}$ $y_i = y_{i-2} - q_i y_{i-1}$

Example: $736x - 464y = 15344$

- Initialize the table (remove the minus in front of b):

i	q_i	r_i	x_i	y_i
-1		736	1	0
0		464	0	1

$$q_i = floor\left(\frac{r_{i-2}}{r_{i-1}}\right)$$

$$r_i = r_{i-2} - q_i r_{i-1}$$

$$x_i = x_{i-2} - q_i x_{i-1}$$

$$y_i = y_{i-2} - q_i y_{i-1}$$

Example: $736x - 464y = 15344$

- Initialize the table (remove the minus in front of b):

i	q_i	r_i	x_i	y_i
-1		736	1	0
0		464	0	1
1	1	272	1	-1

$$q_i = floor\left(\frac{r_{i-2}}{r_{i-1}}\right)$$

$$r_i = r_{i-2} - q_i r_{i-1}$$

$$x_i = x_{i-2} - q_i x_{i-1}$$

$$y_i = y_{i-2} - q_i y_{i-1}$$

$$q_1 = floor\left(\frac{r_{-1}}{r_0}\right) = floor\left(\frac{736}{464}\right) = 1$$

$$x_1 = x_{-1} - q_1 x_0 = 1 - 1 \cdot 0 = 1$$

$$r_1 = r_{-1} - q_1 r_0 = 736 - 1 \cdot 464 = 272$$

$$y_1 = y_{-1} - q_1 y_0 = 0 - 1 \cdot 1 = -1$$

Example: $736x - 464y = 15344$

- Initialize the table (remove the minus in front of b):

i	q_i	r_i	x_i	y_i
-1		736	1	0
0		464	0	1
1	1	272	1	-1
2	1	192	-1	2

$$q_i = floor\left(\frac{r_{i-2}}{r_{i-1}}\right)$$

$$r_i = r_{i-2} - q_i r_{i-1}$$

$$x_i = x_{i-2} - q_i x_{i-1}$$

$$y_i = y_{i-2} - q_i y_{i-1}$$

$$q_2 = floor\left(\frac{r_0}{r_1}\right) = floor\left(\frac{464}{272}\right) = 1$$

$$x_2 = x_0 - q_2 x_1 = 0 - 1 \cdot 1 = -1$$

$$r_2 = r_0 - q_2 r_1 = 464 - 1 \cdot 272 = 192$$

$$y_2 = y_0 - q_2 y_1 = 1 - 1 \cdot (-1) = 2$$

Example: $736x - 464y = 15344$

- Initialize the table (remove the minus in front of b):

i	q_i	r_i	x_i	y_i
-1		736	1	0
0		464	0	1
1	1	272	1	-1
2	1	192	-1	2
3	1	80	2	-3

$$q_i = floor\left(\frac{r_{i-2}}{r_{i-1}}\right)$$

$$r_i = r_{i-2} - q_i r_{i-1}$$

$$x_i = x_{i-2} - q_i x_{i-1}$$

$$y_i = y_{i-2} - q_i y_{i-1}$$

$$q_3 = floor\left(\frac{r_1}{r_2}\right) = floor\left(\frac{272}{192}\right) = 1$$

$$x_3 = x_1 - q_3 x_2 = 1 - 1 \cdot (-1) = 2$$

$$r_3 = r_1 - q_3 r_2 = 272 - 1 \cdot 192 = 80$$

$$y_3 = y_1 - q_3 y_2 = -1 - 1 \cdot 2 = -3$$

Example: $736x - 464y = 15344$

- Initialize the table (remove the minus in front of b):

i	q_i	r_i	x_i	y_i
-1		736	1	0
0		464	0	1
1	1	272	1	-1
2	1	192	-1	2
3	1	80	2	-3
4	2	32	-5	8

$$q_i = floor\left(\frac{r_{i-2}}{r_{i-1}}\right)$$

$$r_i = r_{i-2} - q_i r_{i-1}$$

$$x_i = x_{i-2} - q_i x_{i-1}$$

$$y_i = y_{i-2} - q_i y_{i-1}$$

$$q_4 = floor\left(\frac{r_2}{r_3}\right) = floor\left(\frac{192}{80}\right) = 2$$

$$x_4 = x_2 - q_4 x_3 = -1 - 2 \cdot 2 = -5$$

$$r_4 = r_2 - q_4 r_3 = 192 - 2 \cdot 80 = 32$$

$$y_4 = y_2 - q_4 y_3 = 2 - 2 \cdot (-3) = 8$$

Example: $736x - 464y = 15344$

- Initialize the table (remove the minus in front of b):

i	q_i	r_i	x_i	y_i
-1		736	1	0
0		464	0	1
1	1	272	1	-1
2	1	192	-1	2
3	1	80	2	-3
4	2	32	-5	8
5	2	16	12	-19

$$q_i = floor\left(\frac{r_{i-2}}{r_{i-1}}\right)$$

$$r_i = r_{i-2} - q_i r_{i-1}$$

$$x_i = x_{i-2} - q_i x_{i-1}$$

$$y_i = y_{i-2} - q_i y_{i-1}$$

$$q_5 = floor\left(\frac{r_3}{r_4}\right) = floor\left(\frac{80}{32}\right) = 2$$

$$x_5 = x_3 - q_5 x_4 = 2 - 2 \cdot (-5) = 12$$

$$r_5 = r_3 - q_5 r_4 = 80 - 2 \cdot 32 = 16$$

$$y_5 = y_3 - q_5 y_4 = -3 - 2 \cdot 8 = -19$$

Example: $736x - 464y = 15344$

- Initialize the table (remove the minus in front of b):

i	q_i	r_i	x_i	y_i
-1		736	1	0
0		464	0	1
1	1	272	1	-1
2	1	192	-1	2
3	1	80	2	-3
4	2	32	-5	8
5	2	16	12	-19
6	2	0	-29	46

$$q_i = floor\left(\frac{r_{i-2}}{r_{i-1}}\right)$$

$$r_i = r_{i-2} - q_i r_{i-1}$$

$$x_i = x_{i-2} - q_i x_{i-1}$$

$$y_i = y_{i-2} - q_i y_{i-1}$$

$$q_6 = floor\left(\frac{r_4}{r_5}\right) = floor\left(\frac{32}{16}\right) = 2$$

$$x_6 = x_4 - q_6 x_5 = -5 - 2 \cdot 12 = -29$$

$$r_6 = r_4 - q_6 r_5 = 32 - 2 \cdot 16 = 0$$

$$y_6 = y_4 - q_6 y_5 = 8 - 2 \cdot (-19) = 46$$

(Remainder = 0, so no need to calculate x and y anymore)

Example: $736x - 464y = 15344$

- From the last row where $r_i \neq 0$ we get the solution:
  - $\gcd(736{,}464) = 16, \;\; x_k = 12, \;\; y_k = -19$
- At this point we remember that b was negative, so we have to invert the sign of y_k
- These solutions should now fulfill the equation

$$ax_k + by_k = \gcd(a, b)$$

- Substitute values:

$$736 \cdot 12 - 464 \cdot 19 = 16$$

(Quick calculator check: True!)

- ${}^{15344}/_{16} = 959$, so let's multiply:

$$736 \cdot 12 \cdot 959 - 464 \cdot 19 \cdot 959 = 16 \cdot 959$$

- Now we get the solutions to our original equation:

$$736 \cdot 11508 - 464 \cdot 18221 = 15344 \quad \Rightarrow \quad \begin{cases} \boldsymbol{x = 11\,508} \\ \boldsymbol{y = 18\,221} \end{cases}$$

Prime numbers

- Natural numbers, which are not divisible by any other natural numbers besides 1 and themselves, are called prime numbers
- Prime numbers are of particular interest to mathematicians: they play a very important role in cryptography, blockchains etc.

2	3	5	7	11
13	17	19	23	29
31	37	41	43	47
53	59	61	67	71
73	79	83	89	97

First 25 prime numbers (there are infinitely more, though)

RSA Cryptography

- One system that makes use of all of these is the RSA public-key cryptosystem
  - Named after its inventors (Rivest-Shamir-Adleman)
- RSA relies in the idea that certain operations are easy to perform, but hard to reverse
  - Easy to perform: multiply two large prime numbers $\mathrm{z} = p \cdot q$
  - Hard to perform: factor the z and find out values of p and q
  - Real-life analogy: add salt to soup vs. remove salt from soup
- RSA encryption provides a safe method for sending sensitive information over networks:
  - Credit card numbers
  - Official documents

RSA Cryptography

- How to set up an RSA system:
  1. Choose two large prime numbers p and q
  2. Calculate $z = pq$
  3. Calculate $\phi = (p-1)(q-1)$
  4. Choose n so that $\gcd(n, \phi) = 1$ (hint: choose n prime)
  5. Compute decryption key s so that $ns\ mod\ \phi = 1$ (requires some calculation using the Euclidean algorithm, but not a hard task for a computer)
  6. Publish the public key (z, n) for encryption
- Now everyone who wants to send you secure messages can encrypt their message a using the encryption key:

$$c = a^n mod\ z$$

- You can decrypt the received encrypted message c using the decryption key:

$$a = c^s mod\ \phi$$

Modular exponentiation

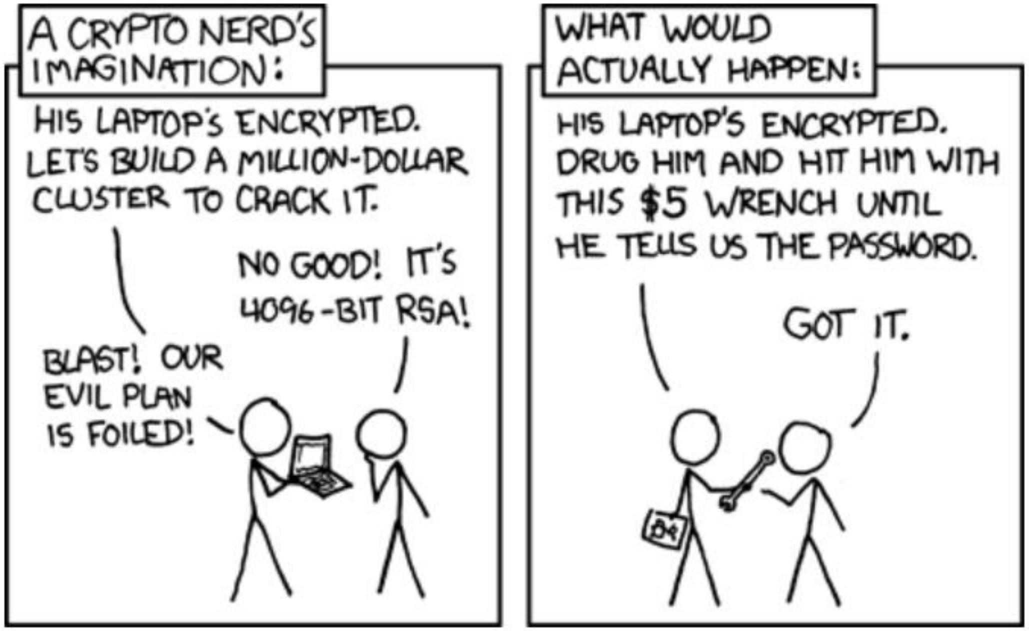
- Numbers where to calculate the modulo from seem huge, so one would think that this calculation takes a long time
- It would, but luckily a technique called modular exponentiation can be used – thanks to this law:

  $$\boldsymbol{ab \bmod z = [(a \bmod z)(b \bmod z)] \bmod z}$$

- Example: $686^{17} \mod 127$
  - Start calculating modulos of powers of 686:
  - $686^2 \mod 127 = 470\,596 \mod 127 = 61$
  - $686^4 \mod 127 = 61^2 \mod 127 = 3721 \mod 127 = 38$
  - $686^8 \mod 127 = 38^2 \mod 127 = 1444 \mod 127 = 47$
  - $686^{16} \mod 127 = 47^2 \mod 127 = 2209 \mod 127 = 50$
  - $686^{17} \mod 127 = 50 \cdot 686 \mod 127 = 34300 \mod 127 = \mathbf{10}$

Safety of cryptography

- Modern cryptography methods (RSA etc.) are reasonably secure – cracking them by pure calculation is almost impossible (in reasonable time)
- However, there are alternatives for number-crunching
- Weakest link in 99 % of cases is the human user


A CRYPTO NERD'S IMAGINATION:
HIS LAPTOP'S ENCRYPTED. LET'S BUILD A MILLION-DOLLAR CLUSTER TO CRACK IT.
NO GOOD! IT'S 4096-BIT RSA!
BLAST! OUR EVIL PLAN IS FOILED!
WHAT WOULD ACTUALLY HAPPEN:
HIS LAPTOP'S ENCRYPTED. DRUG HIM AND HIT HIM WITH THIS $5 WRENCH UNTIL HE TELLS US THE PASSWORD.
GOT IT.

Thank you!