Algorithms and Data Structures
- Lesson 5 -

Michael Schwarzkopf
https://www.uni-weimar.de/de/medien/professuren/medieninformatik/grafische-datenverarbeitung

Bauhaus-University Weimar

June 27, 2018

Overview

...of things you should definetely know about if you want a very good grade

- Fourier Transform (Belongs to Devide & Conquer Lecture)
- Sorting Based
  - Point in Polygon
  - Convex Hull
  - Geometric Cut
- Mathematical
  - All shortest paths: Warshall Algo
  - Gaussian Elimination
  - Random Numbers
- Mathematical: Interpolation and Integration
- NP - Completeness

Fast Fourier Transform

- We are still multipling Polynomials

  Idea:

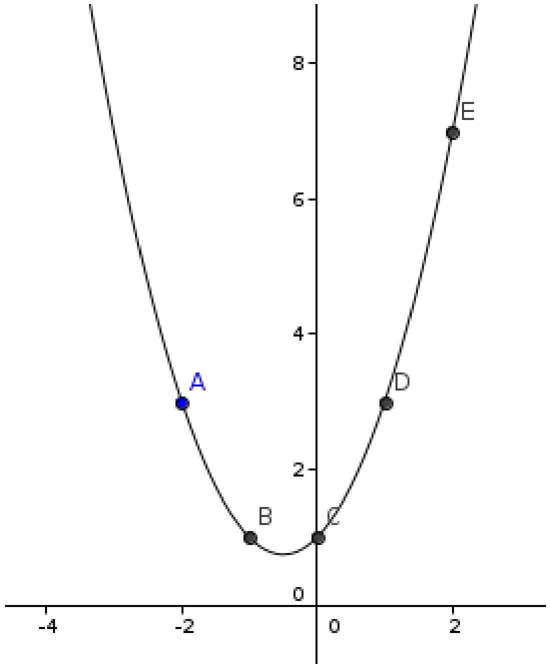
- 2 Polynomials of degree n:
  - Compute 2n - 1 Points of each Polynomial
  - Multiply the y - values
  - Draw Polynomial of degree 2n - 1 through those points

Fast Fourier Transform

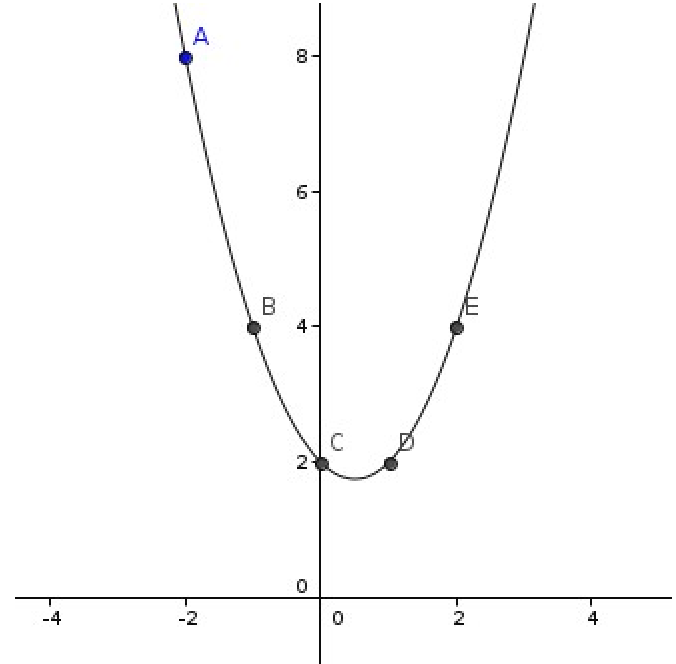
- Lecture: $p(x) = x^2 + x + 1$ $\quad q(x) = x^2 - x + 2$

  $x_i = \{-2, -1, 0, 1, 2\}$ $\leftrightarrow$ $2n - 1 = 5$ Values
- $p(x_i) = \{3, 1, 1, 3, 7\}$ $\quad q(x_i) = \{8, 4, 2, 2, 4\}$
- Visualized:

$p(x)$

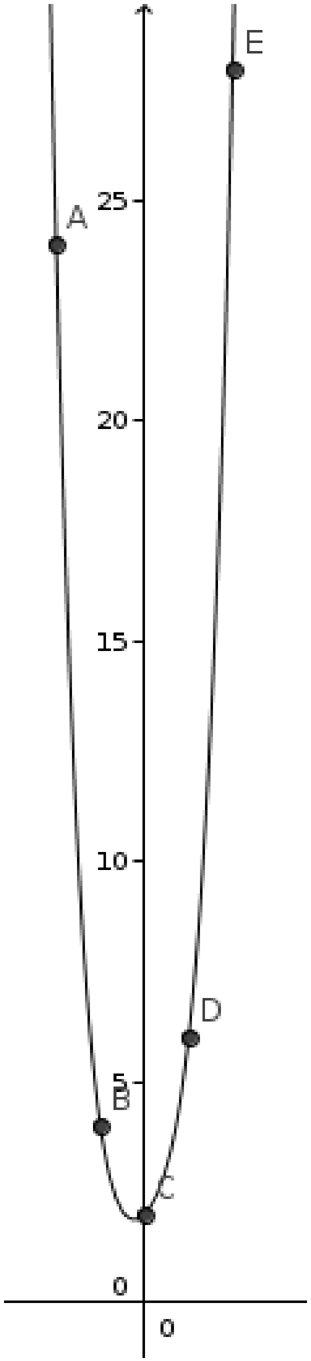


$q(x)$



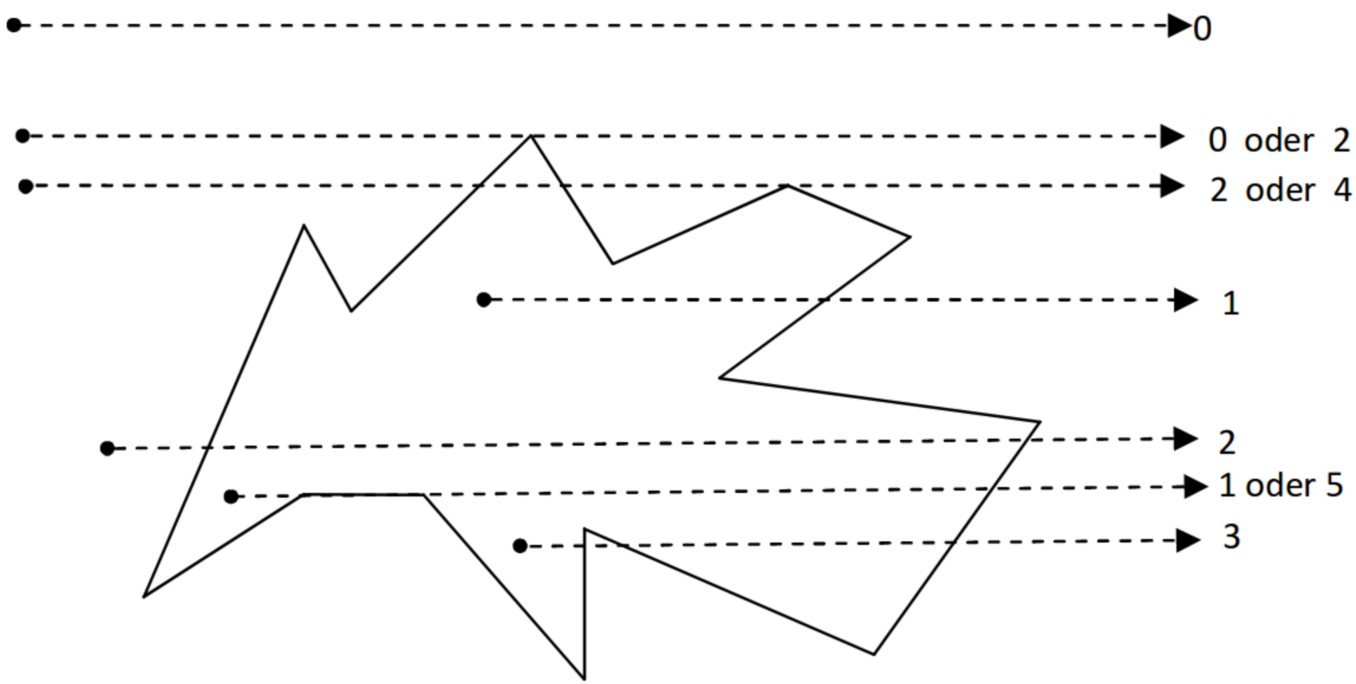
Fast Fourier Transform

- $p(x_i) * q(x_i) = \{24, 4, 2, 6, 28\}$
- Use polynomial interpolaton (e.g. Langrange) to obtain $x^4 + 2x^2 + x + 2$
- If you don't know Lagrange Interpolation by heart, just say „Now I use Interpolation" if somebody asks you.



Point in Polygon

- How to figure out wether a Point is inside of a polygon?



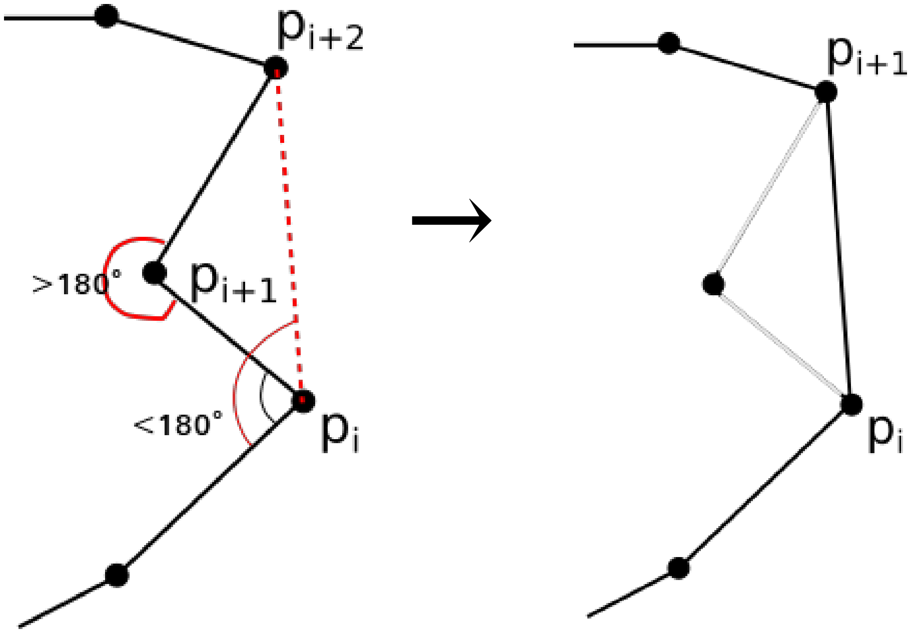
- Number of intersections:

  - Even: Outside

  - Odd: Inside

- Convex hull: „smallest Polygon in which every Point can be connected with another without crossing the line“

Convex hull: Graham's Scan



- Build closed path
  (Lecture, Slide 7)

- Start at p_1:

  if the angle at p_i is bigger than 180°:

    exclude p_i from the Path

    connect p_{i-1} with p_{i+1}, check again at p_{i-1}

  else:

    carry on at p_{i+1}

Convex hull: Wrapping

- Take lowest point (smallest y component)
- While path not closed:
  - Take the point with the smallest angle to the line $y = 0$
  - Connect current point with point of lowest angle
  - Turn the whole set so that last drawn line is horizontal ($y = 0$)

Convex hull: Complexity

- Graham:
  „Build closed path“ = Sort angles.
  Sorting is O(n log n)

  „For each point do…“ → O(n)

  O(n) + O(n log n) = <u>O(n log n)</u> Sorting dominates.

- Wrapping:
  „For each Point in C.H.“ → O(m)
  „Find smallest Point in Set“ → O(n)
  → <u>O(n * m)</u>
  worst case is O(n * n), when each point is in C.H.

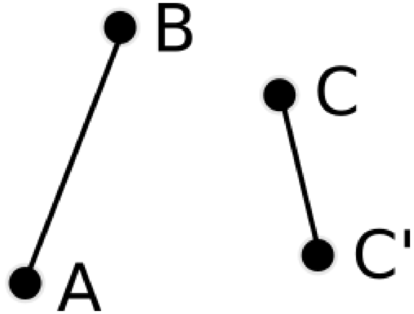
Geometric Cut: Ccw

- Set of lines; We search for all intersections
- For the general Problem we need ccw function (Counterclockwise)
- Ccw: we go from A to B. When we now go to C, do we have to:
  - Turn clockwise: ccw = -1
  - Turn counterclockwise: ccw = 1
  - Don‘t turn at all or 180°: ccw = 0*

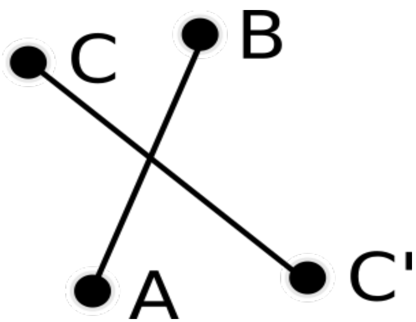
*Differs from a certain general Definition we don‘t need at the moment.

Geometric Cut:

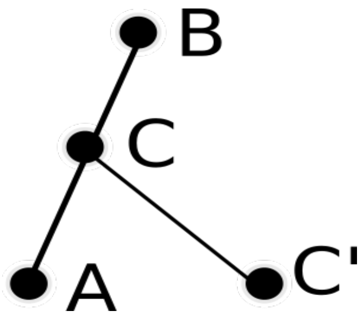
- Line (A, B) , Line (C, C‘):



Ccw(A,B,C) = -1 (C, C‘) is right
Ccw(A,B,C‘) = -1 from (A, B)



Ccw(A,B,C) = 1
Ccw(A,B,C‘) = -1



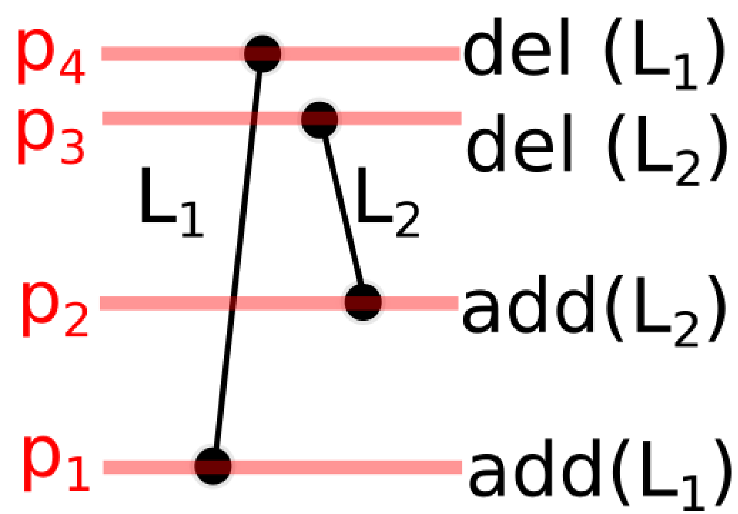
Ccw(A,B,C) = 0
Ccw(A,B,C‘) = -1

- Observe: if Ccw(A,B,C) ≠ Ccw(A,B,C‘): Intersection

Geometric Cut: Bentley-Ottmann

- Sort Points of Line – endings w.r.t. y - components

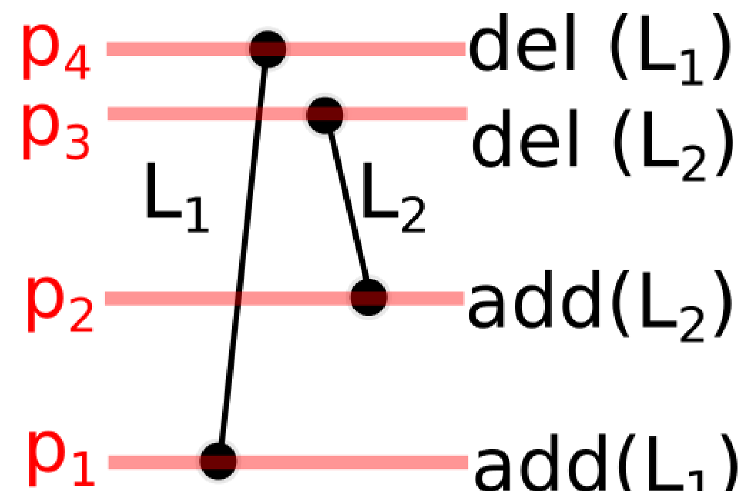
For all p_i in set of points:
    if p_i is a beginning of a line:
        add line to a Datastructure
    if p_i is a ending of a line:
        remove line.



The Datastructure has to provide, that the Points in it are sorted w.r.t. x – component (e.g. like [L_1, L_2] between p_3 and p_4)

Geometric Cut: Bentley-Ottmann

- Do ccw on every Interval $[p_i, p_{i+1}]$
  That's it.



- The best Datastructure is a binary search tree so adding elements takes max. log n steps
- As for each intersection we have to re-execute bintree sort, total complexity is

$$\underline{O((k+n)\log n)}$$

Where k of course is the number of intersections.

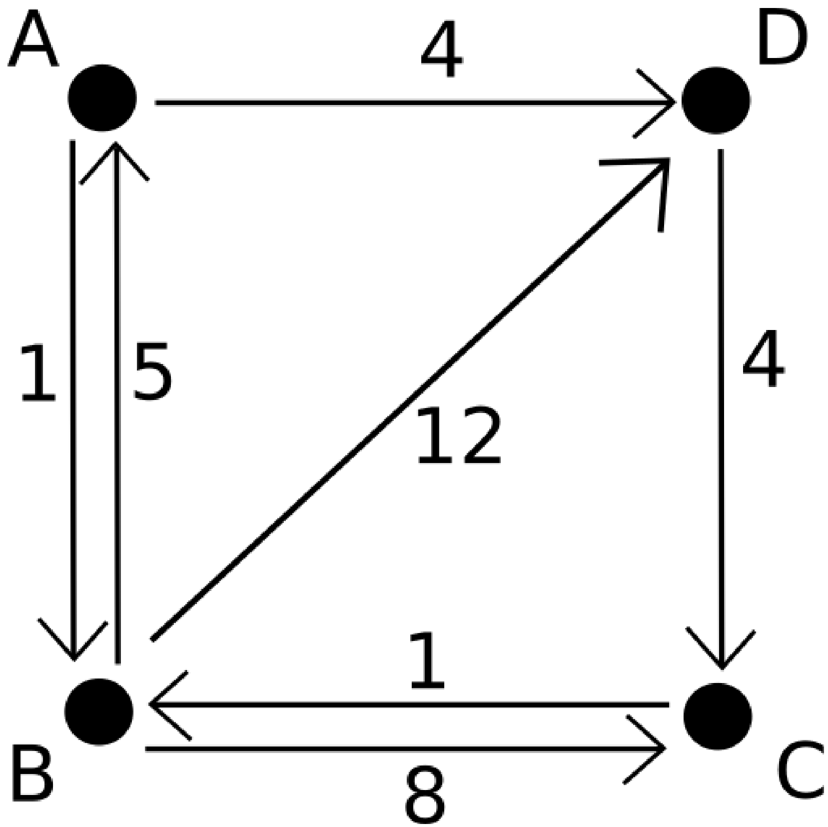
Warshall Algorithm

- How to find all shortest Paths in a digraph?
  - (Or how to find all paths from each vertex, hence transistive hull)
- For all Vertices: Do Dijkstra?
  - Worst Case: $O(V^3 (\log V))$
- Warshall:
  For all Vertices: Check adjacency Matrix

  → $\underline{O(V^3)}$:)

Warshall Algorithm

- Creating Adjecency Matrix:
  - Path from a Vertex to itself is zero
  - Non existing Paths get F for „false“



$$\begin{pmatrix} 0 & 1 & F & 4 \\ 5 & 0 & 8 & 12 \\ F & 1 & 0 & F \\ F & F & 4 & 0 \end{pmatrix}$$

Warshall Algorithm

- For all Vertices k = {1, 2, … , N}

  For all Elements $x \neq k$ and $y \neq k$:

  if $a[x,y] > a[x,k] + a[k,y]$:

  $a[x,y] = a[x,k] + a[k,y]$

$k = 1$

$$\begin{pmatrix} 0 & 1 & F & 4 \\ 5 & 0 & 8 & 12 \\ F & 1 & 0 & F \\ F & F & 4 & 0 \end{pmatrix} \rightarrow \begin{pmatrix} 0 & 1 & F & \underline{4} \\ \underline{5} & 0 & 8 & 9 \\ F & 1 & 0 & F \\ F & F & 4 & 0 \end{pmatrix}$$

- $a[4,2] > a[4,1] + a[1,2]$

  $12 > 4 + 5 \rightarrow a[4,2] = 9$

Warshall Algorithm

- For all Vertices $k = \{1, 2, \ldots, N\}$

  For all Elements $x \neq k$ and $y \neq k$:

  if $a[x,y] > a[x,k] + a[k,y]$:

  $a[x,y] = a[x,k] + a[k,y]$

$k = 2$

$$\begin{pmatrix} 0 & 1 & F & 4 \\ 5 & 0 & 8 & 9 \\ F & 1 & 0 & F \\ F & F & 4 & 0 \end{pmatrix} \rightarrow \begin{pmatrix} 0 & 1 & 9 & 4 \\ 5 & 0 & 8 & 9 \\ 6 & 1 & 0 & 10 \\ F & F & 4 & 0 \end{pmatrix}$$

- We assume, that every number is smaller than F.

$6 = 1 + 5 \qquad 9 = 8 + 1 \qquad 10 = 9 + 1$

Warshall Algorithm

- For all Vertices k = {1, 2, … , N}

  For all Elements $x \neq k$ and $y \neq k$:

  if $a[x,y] > a[x,k] + a[k,y]$:

  $a[x,y] = a[x,k] + a[k,y]$

$k = 3$

$$\begin{pmatrix} \boxed{\begin{matrix} 0 & 1 \\ 5 & 0 \end{matrix}} & 9 & \boxed{\begin{matrix} 4 \\ 9 \end{matrix}} \\ 6 \quad 1 & 0 & 10 \\ \boxed{\begin{matrix} F & F \end{matrix}} & 4 & \boxed{0} \end{pmatrix} \rightarrow \begin{pmatrix} 0 & 1 & 9 & 4 \\ 5 & 0 & 8 & 9 \\ 6 & 1 & 0 & 10 \\ 10 & 5 & 4 & 0 \end{pmatrix}$$

$10 = 6 + 4$ $\qquad$ $5 = 1 + 4$

Warshall Algorithm

- For all Vertices k = {1, 2, … , N}

    For all Elements $x \neq k$ and $y \neq k$:

        if $a[x,y] > a[x,k] + a[k,y]$:

            $a[x,y] = a[x,k] + a[k,y]$

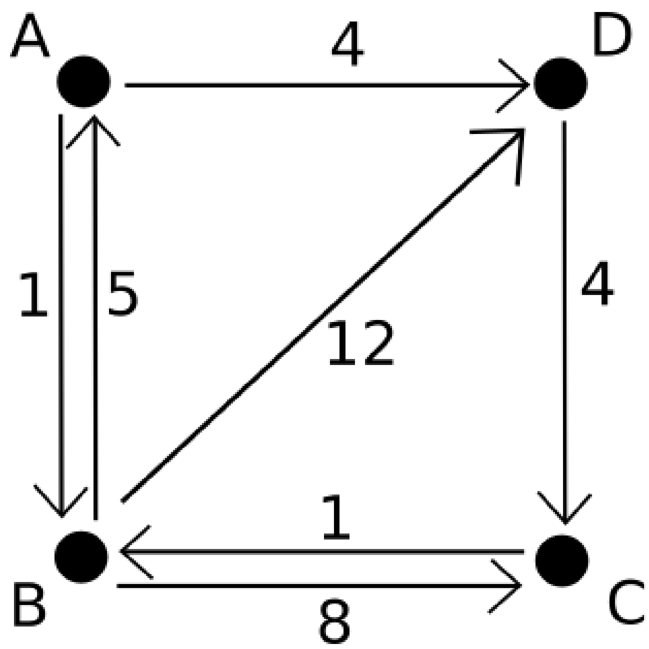
k = 3

$$\begin{pmatrix} 0 & 1 & 9 & 4 \\ 5 & 0 & 8 & 9 \\ 6 & 1 & 0 & 10 \\ 10 & 5 & 4 & 0 \end{pmatrix} \rightarrow \begin{pmatrix} 0 & 1 & 8 & 4 \\ 5 & 0 & 8 & 9 \\ 6 & 1 & 0 & 10 \\ 10 & 5 & 4 & 0 \end{pmatrix}$$

8 = 4 + 4 and we are done.

Warshall Algorithm

- We observe:
  - no F in the Matrix, so every vertex is reachable from each one
  - If we want to know the shortest path from C to D for example, just check a[3,4] = 10



$$\begin{pmatrix} 0 & 1 & 8 & 4 \\ 5 & 0 & 8 & 9 \\ 6 & 1 & 0 & 10 \\ 10 & 5 & 4 & 0 \end{pmatrix}$$

Random Numbers

- Finally let's create some random numbers
- <u>Linear congruence</u>.

  Seed S = 7, modulo m = 13, constant b = 2
- $s_0 = S$, $s_{i+1} = (s_i * b + 1) \bmod M$

$s_0 = 7$

$s_1 = (7 * 2 + 1) \bmod 13 = 2$

$s_2 = (2 * 2 + 1) \bmod 13 = 5$ → 7, 2, 5, 11, 12

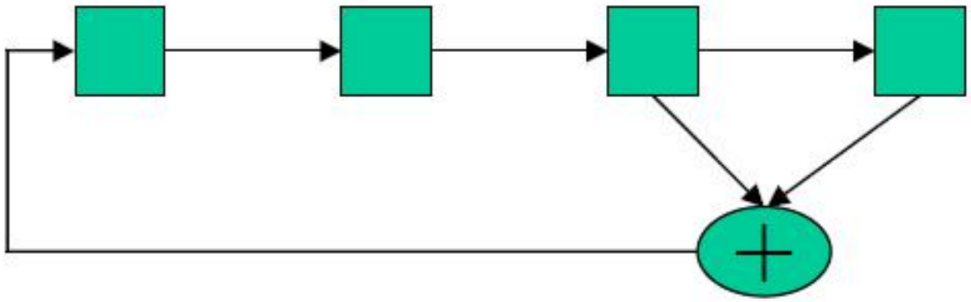
$s_3 = (5 * 2 + 1) \bmod 13 = 11$ So random!

$s_4 = (11 * 2 + 1) \bmod 13 = 12$

Random Numbers

- Additive congruence
  Let's feed our seed into this guy:

$7_{10} = 0111_2$ →

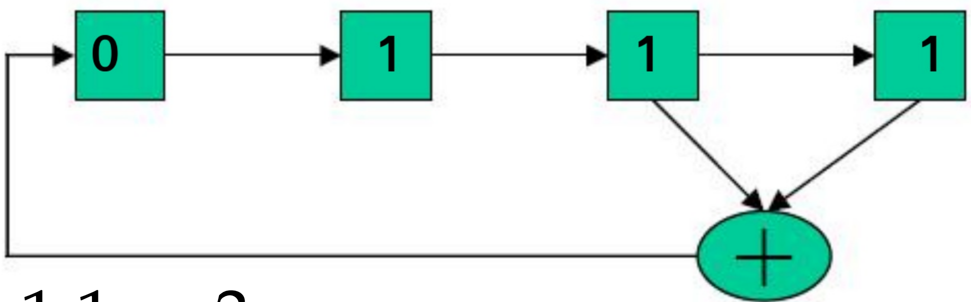


- If the last two digits are 11 or 00, add a 0 at the left
- If they are 01 or 10, add a 1
- After that, delete the right digit
  → shift

Random Numbers

- <u>Additive congruence</u>
  Let's feed our seed into this guy:

$7_{10} = 0111_2 \quad \rightarrow$



$$0\,0\,1\,1 = 3$$

$$0\,0\,0\,1 = 1$$

$$1\,0\,0\,0 = 8$$

$$7, 3, 1, 8, 4, 2, 9 \leftarrow \quad 0\,1\,0\,0 = 4$$

$$0\,0\,1\,0 = 2$$

$$1\,0\,0\,1 = 9$$

Random Numbers

- Be carefull with the Seeds.
  - if you use linear congruence with
    $s = 7, m = 11, b = 13$
    you get 7, 3, 7, 3, 7,…
  - if you use additive congruence with
    0000, you get 0000 again.

- Some of you meight enjoy this stuff in the cryptoraphy course in detail.

Randomness test

- We create N numbers randomly
- Every number is smaller than a certain r
  (Can be archieved through the mod operation for example)
- f_i is just the i th randomly generated number

$$\chi^2 = \frac{\sum_{i=0}^{r}(f_i - N/r)^2}{N/r}$$ ← don't miss this!

- The difference between χ^2 and r is small, the numbers are likely to be random.
  (This means in practice, you should test on very big sets, also more often than once)

Assignment

- Take an Integer as number of points as an Input, create those points on a plane at random Positions.
  Now implement Graham‘s scan which outputs the set of points containing the convex hull!
  Output could look like: (3,1), (4,3), (2,4), (1,3)
- It‘s up to you how you generate random values as well as how you calculate angles.
- Good luck!

Assignment Conditions

- Code comes from nowhere else than your brain!
- .java / .c / .cpp → NO .docx .pdf etc!!
- Good comments make the difference between „alright“ and „very good“!
- Put Matriculation number as comment above
- Deadline: 10 July 2018, 23:59
- Mail: michael.schwarzkopf@uni-weimar.de