Fast arithmetic on Hessian curves

Chitchanok Chuengsatiansup

Technische Universiteit Eindhoven

September 5th, 2016

Fast arithmetic on Hessian curves

Chitchanok Chuengsatiansup

Technische Universiteit Eindhoven

September 5th, 2016

Joint work with Daniel J. Bernstein, David Kohel and Tanja Lange

Curve shapes

- Short Weierstrass curves
  $y^2 = x^3 + ax + b$

https://hyperelliptic.org/EFD/

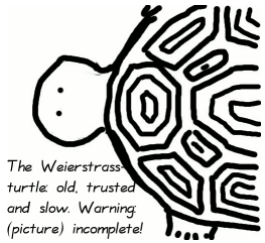
Curve shapes

- Short Weierstrass curves
  $y^2 = x^3 + ax + b$
  e.g., $y^2 = x^3 - 0.4x + 7$

https://hyperelliptic.org/EFD/

Curve shapes

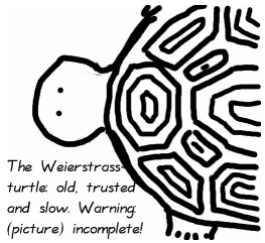
- Short Weierstrass curves
  $y^2 = x^3 + ax + b$
  e.g., $y^2 = x^3 - 0.4x + 7$



https://hyperelliptic.org/EFD/

Curve shapes

- Short Weierstrass curves
  $y^2 = x^3 + ax + b$
  e.g., $y^2 = x^3 - 0.4x + 7$

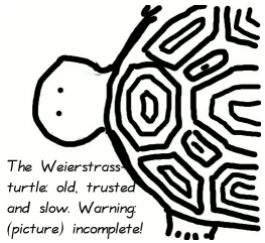


- Edwards curves
  $x^2 + y^2 = 1 + dx^2y^2$

https://hyperelliptic.org/EFD/

Curve shapes

- Short Weierstrass curves
  $y^2 = x^3 + ax + b$
  e.g., $y^2 = x^3 - 0.4x + 7$

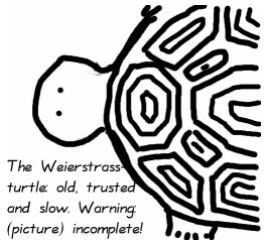


- Edwards curves
  $x^2 + y^2 = 1 + dx^2y^2$
  e.g., $x^2 + y^2 = 1 - 300x^2y^2$

https://hyperelliptic.org/EFD/

Curve shapes

- Short Weierstrass curves
  $y^2 = x^3 + ax + b$
  e.g., $y^2 = x^3 - 0.4x + 7$

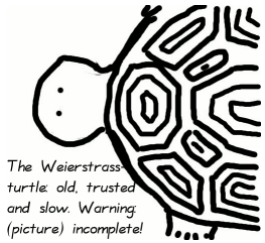


- Edwards curves
  $x^2 + y^2 = 1 + dx^2y^2$
  e.g., $x^2 + y^2 = 1 - 300x^2y^2$

https://hyperelliptic.org/EFD/

Curve shapes

- Short Weierstrass curves
  $y^2 = x^3 + ax + b$
  e.g., $y^2 = x^3 - 0.4x + 7$



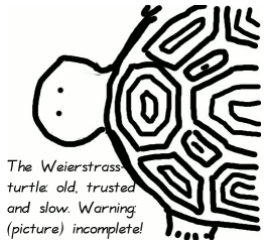
- Edwards curves
  $x^2 + y^2 = 1 + dx^2y^2$
  e.g., $x^2 + y^2 = 1 - 300x^2y^2$

- Hessian curves
  $x^3 + y^3 + 1 = dxy$

https://hyperelliptic.org/EFD/

Curve shapes

- Short Weierstrass curves
  $y^2 = x^3 + ax + b$
  e.g., $y^2 = x^3 - 0.4x + 7$



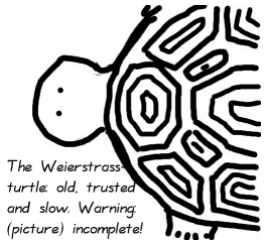
- Edwards curves
  $x^2 + y^2 = 1 + dx^2y^2$
  e.g., $x^2 + y^2 = 1 - 300x^2y^2$

- Hessian curves
  $x^3 + y^3 + 1 = dxy$
  e.g. $x^3 - y^3 + 1 = 0.3xy$

https://hyperelliptic.org/EFD/

Curve shapes

- Short Weierstrass curves
  $y^2 = x^3 + ax + b$
  e.g., $y^2 = x^3 - 0.4x + 7$



- Edwards curves
  $x^2 + y^2 = 1 + dx^2y^2$
  e.g., $x^2 + y^2 = 1 - 300x^2y^2$

- Hessian curves
  $x^3 + y^3 + 1 = dxy$
  e.g. $x^3 - y^3 + 1 = 0.3xy$

https://hyperelliptic.org/EFD/

Cofactors

cofactor $h = \#E/r$ where r is prime

- Short Weierstrass curves
  $y^2 = x^3 + ax + b$
- Edwards curves
  $x^2 + y^2 = 1 + dx^2y^2$
- Hessian curves
  $x^3 + y^3 + 1 = dxy$

Cofactors

cofactor $h = \#E/r$ where r is prime

- Short Weierstrass curves
  $y^2 = x^3 + ax + b$
  cofactor divisible by 1

  NIST's standard prime-field curves

- Edwards curves
  $x^2 + y^2 = 1 + dx^2y^2$

- Hessian curves
  $x^3 + y^3 + 1 = dxy$

Cofactors

cofactor $h = \#E/r$ where r is prime

- Short Weierstrass curves
  $y^2 = x^3 + ax + b$
  cofactor divisible by 1
  NIST's standard prime-field curves
- Edwards curves
  $x^2 + y^2 = 1 + dx^2y^2$
  cofactor divisible by 4
  set many speed records
- Hessian curves
  $x^3 + y^3 + 1 = dxy$

Cofactors

cofactor $h = \#E/r$ where r is prime

- Short Weierstrass curves
  $y^2 = x^3 + ax + b$
  cofactor divisible by 1

  NIST's standard prime-field curves

- Edwards curves
  $x^2 + y^2 = 1 + dx^2y^2$
  cofactor divisible by 4

  set many speed records

- Hessian curves
  $x^3 + y^3 + 1 = dxy$
  cofactor divisible by 3

  classic view: slower than Weierstrass
  new: faster than Weierstrass

Point operation costs

Curve shape	Addition	Doubling	Tripling
Short Weierstrass $y^2 = x^3 - 3x + b$	$11\mathbf{M} + 5\mathbf{S}$	$3\mathbf{M} + 5\mathbf{S}$	$7\mathbf{M} + 7\mathbf{S}$
Doche-Icart-Kohel $y^2 = x^3 + 3c(x+1)^2$	$11\mathbf{M} + 6\mathbf{S}$	$2\mathbf{M} + 7\mathbf{S}$	$6\mathbf{M} + 6\mathbf{S}$
Twisted Hessian $ax^3 + y^3 + 1 = dxy$	$11\mathbf{M}$	$6\mathbf{M} + 2\mathbf{S}$	$6\mathbf{M} + 6\mathbf{S}$

https://hyperelliptic.org/EFD/

Cost comparisons

Cost per bit to compute scalar multiplication for 256-bit (assume $\mathbf{S} = 0.8\mathbf{M}$)

Base	Short Weierstrass	Twisted Hessian	Remark
single	9.85**M**	10.54**M**	Hisil
single	9.34**M**	9.93**M**	BL
double	9.29**M**	9.65**M**	BBLP
double	-	8.77**M**	BCKL (new)

Cost comparisons

Cost per bit to compute scalar multiplication for 256-bit (assume $\mathbf{S} = 0.8\mathbf{M}$)

Base	Short Weierstrass	Twisted Hessian	Remark
single	9.85**M**	10.54**M**	Hisil
single	9.34**M**	9.93**M**	BL
double	9.29**M**	9.65**M**	BBLP
double	-	8.77**M**	BCKL (new)
double	9.13**M**	8.52**M**	newer

Methods

- Improve point formulas
  - tripling
    $$8\mathbf{M} + 6\mathbf{S} \quad \rightarrow \quad 6\mathbf{M} + 6\mathbf{S}$$
  - doubling
    $$7\mathbf{M} + 1\mathbf{S} \quad \rightarrow \quad 6\mathbf{M} + 2\mathbf{S}$$
  - see http://cr.yp.to/papers.html#hessian

Methods

- Improve point formulas
  - tripling
    $$8\mathbf{M} + 6\mathbf{S} \quad \rightarrow \quad 6\mathbf{M} + 6\mathbf{S}$$
  - doubling
    $$7\mathbf{M} + 1\mathbf{S} \quad \rightarrow \quad 6\mathbf{M} + 2\mathbf{S}$$
  - see http://cr.yp.to/papers.html#hessian

- Generalize Doche–Habsieger double-base chains
  - allow $n \pm c$ where c is from precomputed set
  - continue searching even after finding the first chain
  - weight each node with operation cost

Faster point arithmetic on Hessian curves

Standard addition law

- Let H be the twisted Hessian curve
  $aX^3 + Y^3 + Z^3 = dXYZ$ over a field k

Standard addition law

- Let H be the twisted Hessian curve
  $aX^3 + Y^3 + Z^3 = dXYZ$ over a field k
- Let $(X_1 : Y_1 : Z_1)$ and $(X_2 : Y_2 : Z_2)$ be points on H

Standard addition law

- Let H be the twisted Hessian curve
  $aX^3 + Y^3 + Z^3 = dXYZ$ over a field k
- Let $(X_1 : Y_1 : Z_1)$ and $(X_2 : Y_2 : Z_2)$ be points on H
- Define
  $$X_3 = X_1^2 Y_2 Z_2 - X_2^2 Y_1 Z_1$$
  $$Y_3 = Z_1^2 X_2 Y_2 - Z_2^2 X_1 Y_1$$
  $$Z_3 = Y_1^2 X_2 Z_2 - Y_2^2 X_1 Z_1$$

Standard addition law

- Let H be the twisted Hessian curve
  $aX^3 + Y^3 + Z^3 = dXYZ$ over a field k
- Let $(X_1 : Y_1 : Z_1)$ and $(X_2 : Y_2 : Z_2)$ be points on H
- Define
  $$X_3 = X_1^2 Y_2 Z_2 - X_2^2 Y_1 Z_1$$
  $$Y_3 = Z_1^2 X_2 Y_2 - Z_2^2 X_1 Y_1$$
  $$Z_3 = Y_1^2 X_2 Z_2 - Y_2^2 X_1 Z_1$$
- If $(X_3, Y_3, Z_3) \neq (0, 0, 0)$ then
  $(X_1 : Y_1 : Z_1) + (X_2 : Y_2 : Z_2) = (X_3 : Y_3 : Z_3)$

Standard addition law

- Let H be the twisted Hessian curve
  $aX^3 + Y^3 + Z^3 = dXYZ$ over a field k
- Let $(X_1 : Y_1 : Z_1)$ and $(X_2 : Y_2 : Z_2)$ be points on H
- Define
  $$X_3 = X_1^2 Y_2 Z_2 - X_2^2 Y_1 Z_1$$
  $$Y_3 = Z_1^2 X_2 Y_2 - Z_2^2 X_1 Y_1$$
  $$Z_3 = Y_1^2 X_2 Z_2 - Y_2^2 X_1 Z_1$$
- If $(X_3, Y_3, Z_3) \neq (0, 0, 0)$ then
  $(X_1 : Y_1 : Z_1) + (X_2 : Y_2 : Z_2) = (X_3 : Y_3 : Z_3)$
- When would $(X_3, Y_3, Z_3) = (0, 0, 0)$?
- What to do if $(X_3, Y_3, Z_3) = (0, 0, 0)$?

Standard addition law

- $(X_3, Y_3, Z_3) = (0, 0, 0)$ *if and only if* $(X_2 : Y_2 : Z_2) = (\omega^2 X_1 : \omega Y_1 : Z_1)$ for some $\omega \in k$ with $\omega^3 = 1$

  recall

$$X_3 = X_1^2 Y_2 Z_2 - X_2^2 Y_1 Z_1$$
$$Y_3 = Z_1^2 X_2 Y_2 - Z_2^2 X_1 Y_1$$
$$Z_3 = Y_1^2 X_2 Z_2 - Y_2^2 X_1 Z_1$$

Standard addition law

- $(X_3, Y_3, Z_3) = (0, 0, 0)$ *if and only if*
  $(X_2 : Y_2 : Z_2) = (\omega^2 X_1 : \omega Y_1 : Z_1)$
  for some $\omega \in k$ with $\omega^3 = 1$
  recall
  $$X_3 = X_1^2 Y_2 Z_2 - X_2^2 Y_1 Z_1$$
  $$Y_3 = Z_1^2 X_2 Y_2 - Z_2^2 X_1 Y_1$$
  $$Z_3 = Y_1^2 X_2 Z_2 - Y_2^2 X_1 Z_1$$
- If addition law fails, i.e., $(X_3, Y_3, Z_3) = (0, 0, 0)$
  use other formulas

Standard addition law

- $(X_3, Y_3, Z_3) = (0, 0, 0)$ *if and only if*
  $(X_2 : Y_2 : Z_2) = (\omega^2 X_1 : \omega Y_1 : Z_1)$
  for some $\omega \in k$ with $\omega^3 = 1$
  recall
  $$X_3 = X_1^2 Y_2 Z_2 - X_2^2 Y_1 Z_1$$
  $$Y_3 = Z_1^2 X_2 Y_2 - Z_2^2 X_1 Y_1$$
  $$Z_3 = Y_1^2 X_2 Z_2 - Y_2^2 X_1 Z_1$$
- If addition law fails, i.e., $(X_3, Y_3, Z_3) = (0, 0, 0)$
  use other formulas
- Due to exceptional cases:
  - formulas are incomplete
  - implementors have to check for these conditions

Rotated addition law

- Let H be the twisted Hessian curve
  $aX^3 + Y^3 + Z^3 = dXYZ$ over a field k

Rotated addition law

- Let H be the twisted Hessian curve
  $aX^3 + Y^3 + Z^3 = dXYZ$ over a field k
- Assume that $c \in \overline{k}$ satisfies $c^3 = a$, then
  $(1 : -c : 0) \in H(\overline{k})$

Rotated addition law

- Let H be the twisted Hessian curve
  $aX^3 + Y^3 + Z^3 = dXYZ$ over a field k
- Assume that $c \in \overline{k}$ satisfies $c^3 = a$, then
  $(1 : -c : 0) \in H(\overline{k})$
- If $(X_1 : Y_1 : Z_1) \in H(\overline{k})$, then
  $(X_1 : Y_1 : Z_1) + (1 : -c : 0) = (Y_1 : cZ_1 : c^2X_1)$

Rotated addition law

The rotated addition law is obtained as follows

1. subtract $(1 : -c : 0)$ from one input
   where c is a cube root of a
   recall $aX^3 + Y^3 + Z^3 = dXYZ$
2. use standard addition law
3. add $(1 : -c : 0)$ to output

Rotated addition law

- Let H be the twisted Hessian curve
  $aX^3 + Y^3 + Z^3 = dXYZ$ over a field k

Rotated addition law

- Let H be the twisted Hessian curve
  $aX^3 + Y^3 + Z^3 = dXYZ$ over a field k
- Let $(X_1 : Y_1 : Z_1)$ and $(X_2 : Y_2 : Z_2)$ be points on H

Rotated addition law

- Let H be the twisted Hessian curve
  $aX^3 + Y^3 + Z^3 = dXYZ$ over a field k
- Let $(X_1 : Y_1 : Z_1)$ and $(X_2 : Y_2 : Z_2)$ be points on H
- Define

$$\begin{aligned} X_3' &= Z_2^2 X_1 Z_1 - Y_1^2 X_2 Y_2 \\ Y_3' &= Y_2^2 Y_1 Z_1 - aX_1^2 X_2 Z_2 \\ Z_3' &= aX_2^2 X_1 Y_1 - Z_1^2 Y_2 Z_2 \end{aligned}$$

Rotated addition law

- Let H be the twisted Hessian curve
  $aX^3 + Y^3 + Z^3 = dXYZ$ over a field k
- Let $(X_1 : Y_1 : Z_1)$ and $(X_2 : Y_2 : Z_2)$ be points on H
- Define
  $$\begin{aligned} X_3' &= Z_2^2 X_1 Z_1 - Y_1^2 X_2 Y_2 \\ Y_3' &= Y_2^2 Y_1 Z_1 - aX_1^2 X_2 Z_2 \\ Z_3' &= aX_2^2 X_1 Y_1 - Z_1^2 Y_2 Z_2 \end{aligned}$$
- If $(X_3', Y_3', Z_3') \neq (0, 0, 0)$ then
  $(X_1 : Y_1 : Z_1) + (X_2 : Y_2 : Z_2) = (X_3' : Y_3' : Z_3')$

Rotated addition law

- Let H be the twisted Hessian curve
  $aX^3 + Y^3 + Z^3 = dXYZ$ over a field k
- Let $(X_1 : Y_1 : Z_1)$ and $(X_2 : Y_2 : Z_2)$ be points on H
- Define
$$\begin{aligned} X_3' &= Z_2^2 X_1 Z_1 - Y_1^2 X_2 Y_2 \\ Y_3' &= Y_2^2 Y_1 Z_1 - aX_1^2 X_2 Z_2 \\ Z_3' &= aX_2^2 X_1 Y_1 - Z_1^2 Y_2 Z_2 \end{aligned}$$
- If $(X_3', Y_3', Z_3') \neq (0, 0, 0)$ then
  $(X_1 : Y_1 : Z_1) + (X_2 : Y_2 : Z_2) = (X_3' : Y_3' : Z_3')$

- When would $(X_3', Y_3', Z_3') = (0, 0, 0)$?
- What to do if $(X_3', Y_3', Z_3') = (0, 0, 0)$?

Rotated addition law

- $(X_3', Y_3', Z_3') = (0, 0, 0)$ if and only if
  $(X_2 : Y_2 : Z_2) = (Z_1 : \gamma^2 X_1 : \gamma Y_1)$
  for some $\gamma \in k$ with $\gamma^3 = a$

Rotated addition law

- $(X_3', Y_3', Z_3') = (0, 0, 0)$ if and only if
  $(X_2 : Y_2 : Z_2) = (Z_1 : \gamma^2 X_1 : \gamma Y_1)$
  for some $\gamma \in k$ with $\gamma^3 = a$
- $(X_3', Y_3', Z_3') \neq (0, 0, 0)$
  if $(X_2 : Y_2 : Z_2) = (X_1 : Y_1 : Z_1)$

Rotated addition law

- $(X_3', Y_3', Z_3') = (0, 0, 0)$ if and only if
  $(X_2 : Y_2 : Z_2) = (Z_1 : \gamma^2 X_1 : \gamma Y_1)$
  for some $\gamma \in k$ with $\gamma^3 = a$
- $(X_3', Y_3', Z_3') \neq (0, 0, 0)$
  if $(X_2 : Y_2 : Z_2) = (X_1 : Y_1 : Z_1)$
  this works fine for doubling!

Rotated addition law

- $(X_3', Y_3', Z_3') = (0, 0, 0)$ if and only if
  $(X_2 : Y_2 : Z_2) = (Z_1 : \gamma^2 X_1 : \gamma Y_1)$
  for some $\gamma \in k$ with $\gamma^3 = a$
- $(X_3', Y_3', Z_3') \neq (0, 0, 0)$
  if $(X_2 : Y_2 : Z_2) = (X_1 : Y_1 : Z_1)$
  this works fine for doubling!
- If a is not a cube in k, then
  $(X_3', Y_3', Z_3') \neq (0, 0, 0)$ and
  $(X_1 : Y_1 : Z_1) + (X_2 : Y_2 : Z_2) = (X_3' : Y_3' : Z_3')$

Rotated addition law

- $(X_3', Y_3', Z_3') = (0, 0, 0)$ if and only if
  $(X_2 : Y_2 : Z_2) = (Z_1 : \gamma^2 X_1 : \gamma Y_1)$
  for some $\gamma \in k$ with $\gamma^3 = a$
- $(X_3', Y_3', Z_3') \neq (0, 0, 0)$
  if $(X_2 : Y_2 : Z_2) = (X_1 : Y_1 : Z_1)$
  this works fine for doubling!
- If a is not a cube in k, then
  $(X_3', Y_3', Z_3') \neq (0, 0, 0)$ and
  $(X_1 : Y_1 : Z_1) + (X_2 : Y_2 : Z_2) = (X_3' : Y_3' : Z_3')$
  formulas are complete!

Completeness

- Let H be the twisted Hessian curve
  $aX^3 + Y^3 + Z^3 = dXYZ$ over a field k

Completeness

- Let H be the twisted Hessian curve
  $aX^3 + Y^3 + Z^3 = dXYZ$ over a field k
- Let $(X_1 : Y_1 : Z_1)$ and $(X_2 : Y_2 : Z_2)$ be points on H

Completeness

- Let H be the twisted Hessian curve
  $aX^3 + Y^3 + Z^3 = dXYZ$ over a field k
- Let $(X_1 : Y_1 : Z_1)$ and $(X_2 : Y_2 : Z_2)$ be points on H
- Define (X_3, Y_3, Z_3) and (X_3', Y_3', Z_3') as previous, then
  $(X_3, Y_3, Z_3) \neq (0, 0, 0)$ or $(X_3', Y_3', Z_3') \neq (0, 0, 0)$
  standard add and rotated add form **complete** system

Completeness

- Let H be the twisted Hessian curve
  $aX^3 + Y^3 + Z^3 = dXYZ$ over a field k
- Let $(X_1 : Y_1 : Z_1)$ and $(X_2 : Y_2 : Z_2)$ be points on H
- Define (X_3, Y_3, Z_3) and (X_3', Y_3', Z_3') as previous, then
  $(X_3, Y_3, Z_3) \neq (0, 0, 0)$ or $(X_3', Y_3', Z_3') \neq (0, 0, 0)$
  standard add and rotated add form **complete** system
- If a is not a cube, then
  the rotated addition law **by itself** is complete

Double-base chains

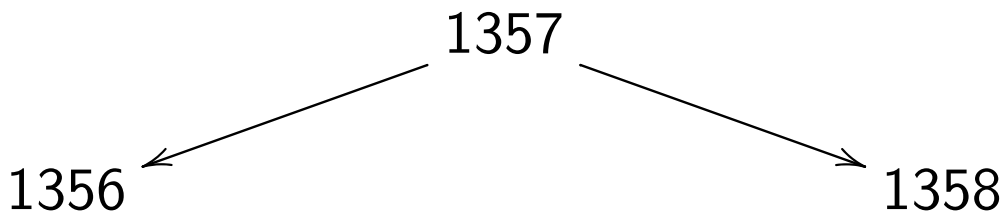
Tree search

(Doche and Habsieger 2008)

1357

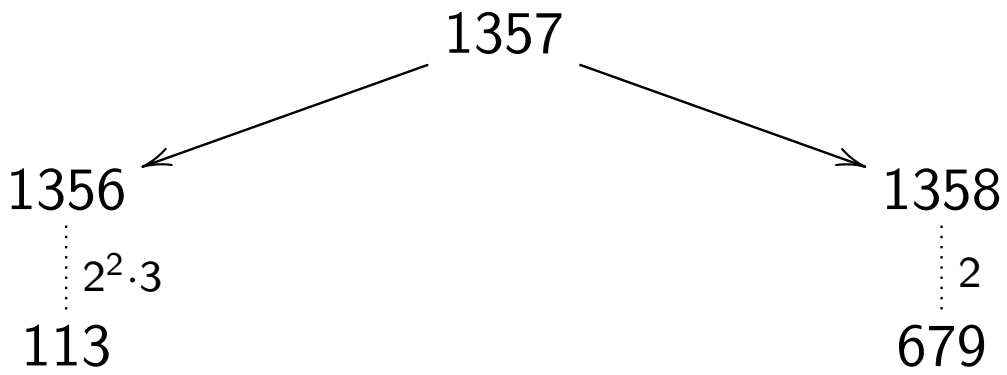
Tree search

(Doche and Habsieger 2008)



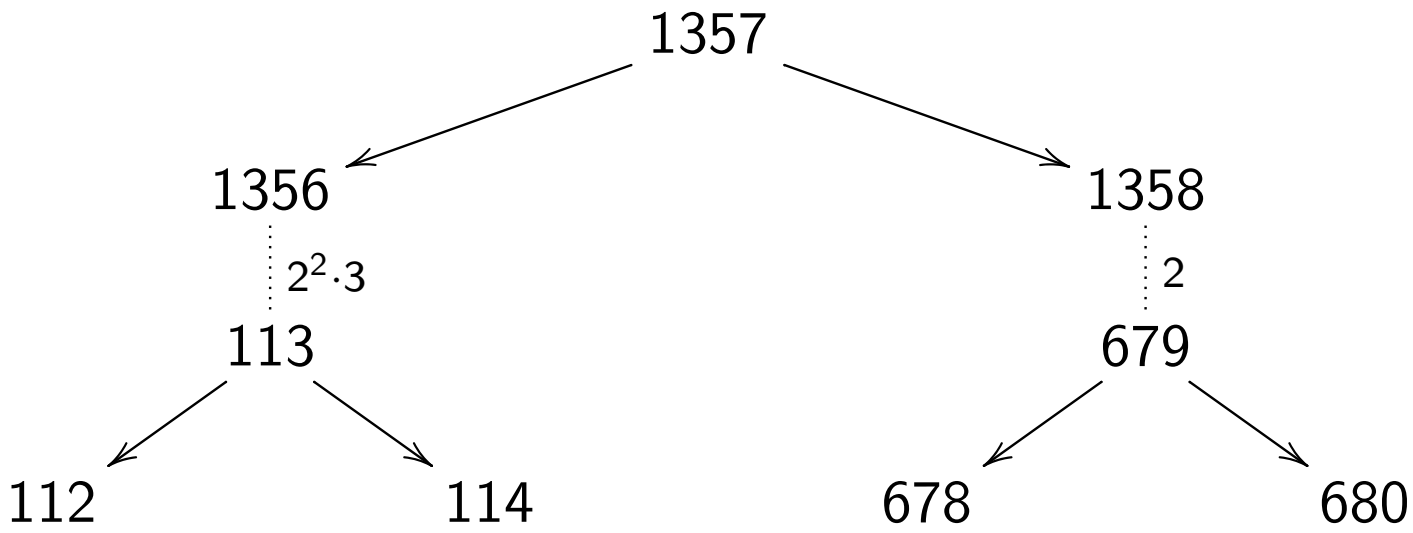
Tree search

(Doche and Habsieger 2008)



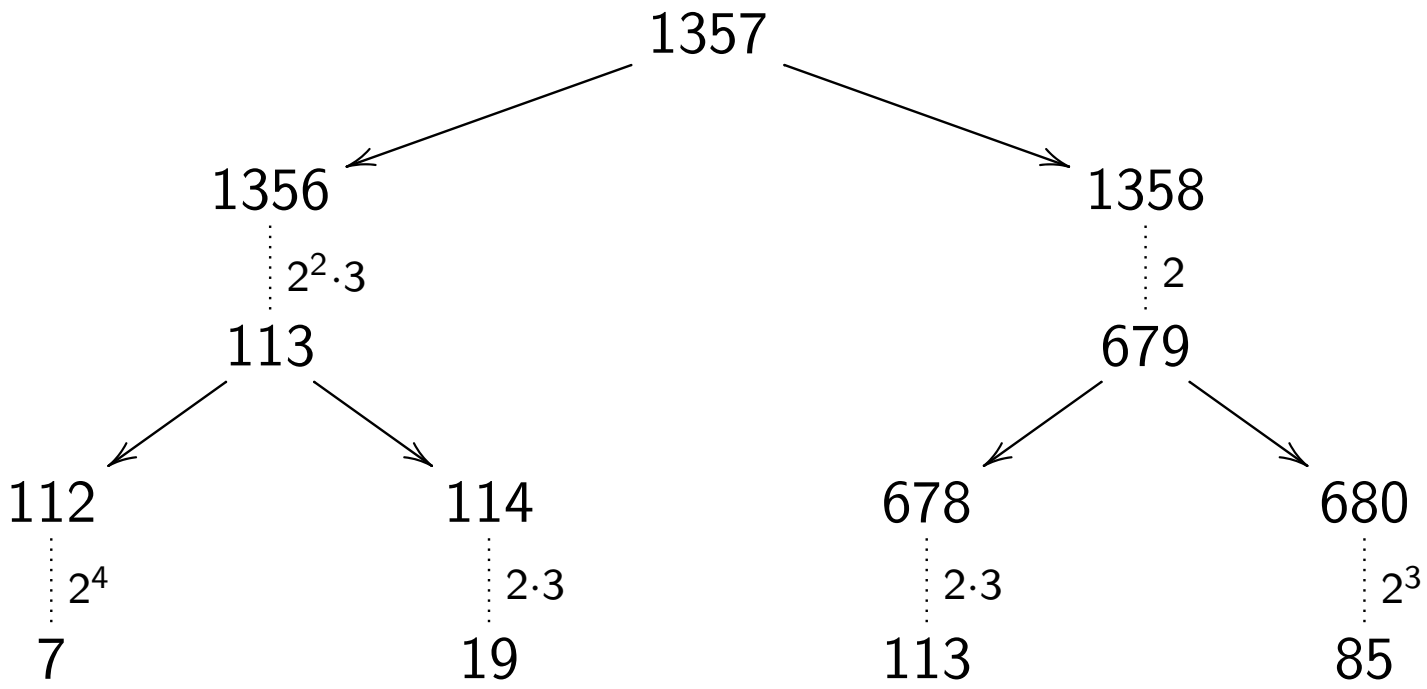
Tree search

(Doche and Habsieger 2008)



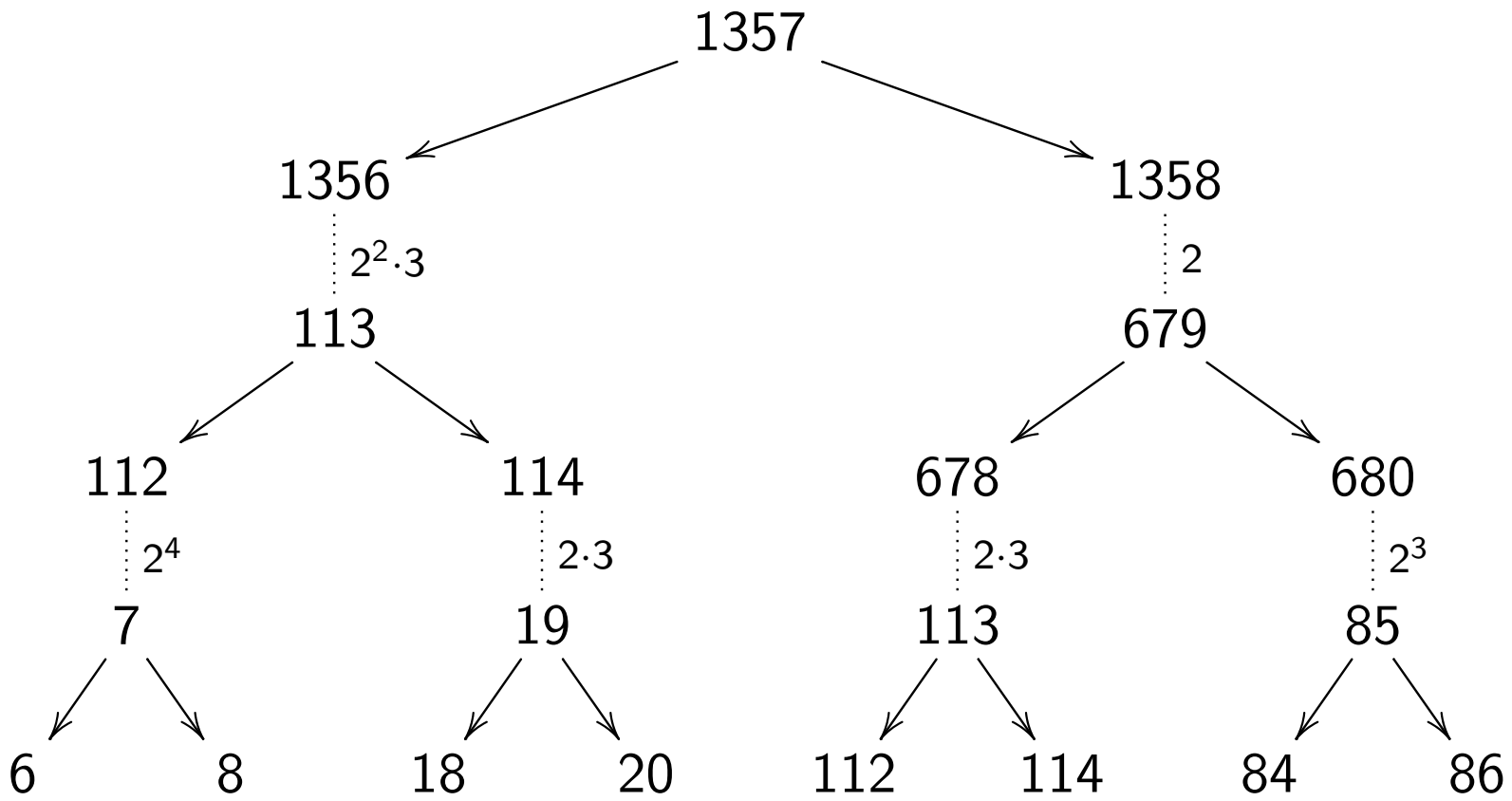
Tree search

(Doche and Habsieger 2008)



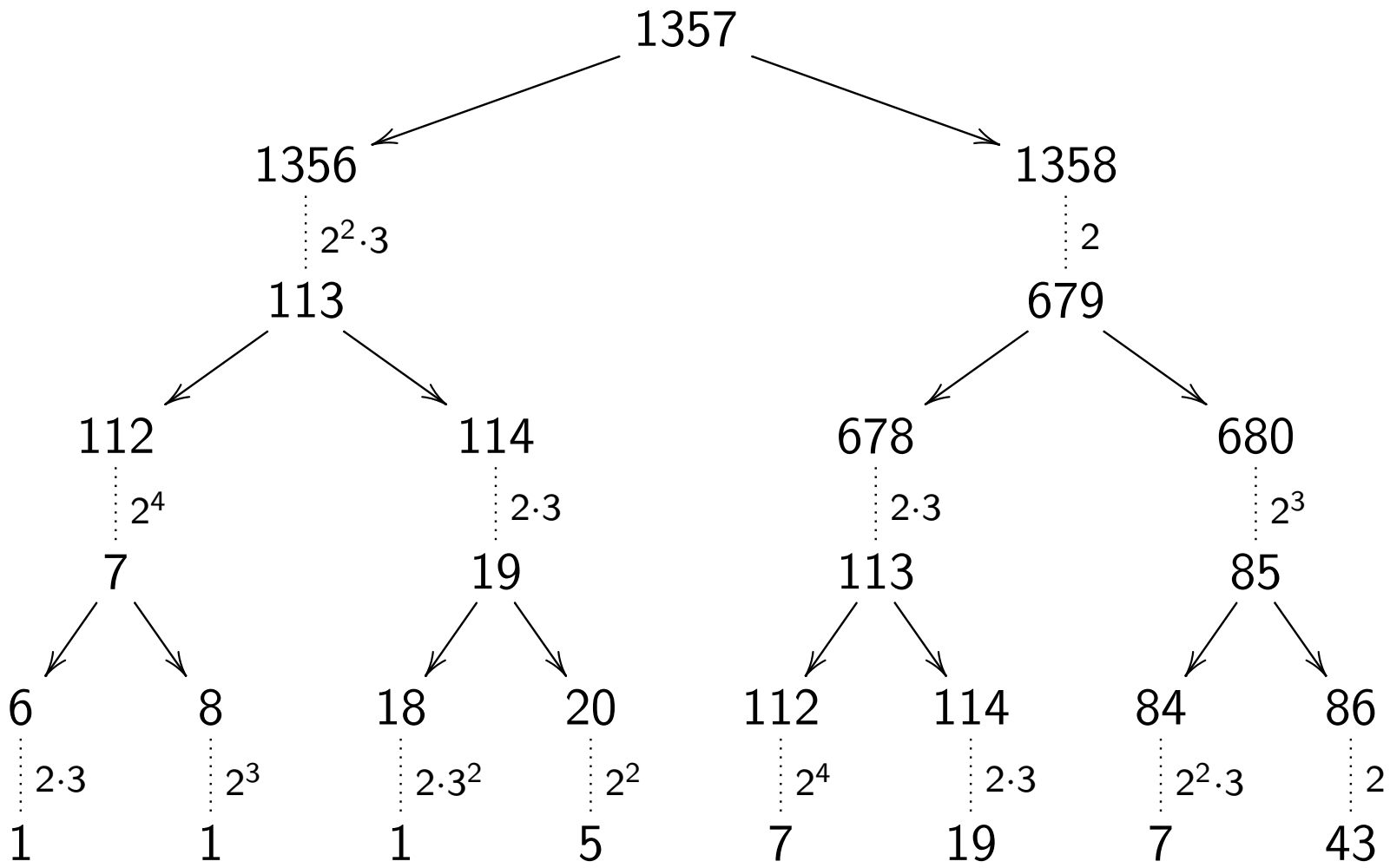
Tree search

(Doche and Habsieger 2008)



Tree search

(Doche and Habsieger 2008)



Our algorithm

We improved "tree-based" approach as follows:

Our algorithm

We improved "tree-based" approach as follows:

- use not just $n-1$ and $n+1$, but all $n-c$
  where c is in a precomputed set S (including both $\pm$ values)

Our algorithm

We improved "tree-based" approach as follows:

- use not just $n-1$ and $n+1$, but all $n-c$ where c is in a precomputed set S (including both $\pm$ values)
- add new nodes to n as follows:
  - **one** child node $n/2$ if n divisible by 2
  - **one** child node $n/3$ if n divisible by 3
  - **several** child nodes $n-c$, one for each $c \in S$

Our algorithm

We improved "tree-based" approach as follows:

- use not just $n-1$ and $n+1$, but all $n-c$
  where c is in a precomputed set S (including both $\pm$ values)
- add new nodes to n as follows:
  - **one** child node $n/2$ if n divisible by 2
  - **one** child node $n/3$ if n divisible by 3
  - **several** child nodes $n-c$, one for each $c \in S$
- continue searching until C chains are found
  rather than stopping with the first chain
  (use $C = 200$, choose the lowest-cost one)

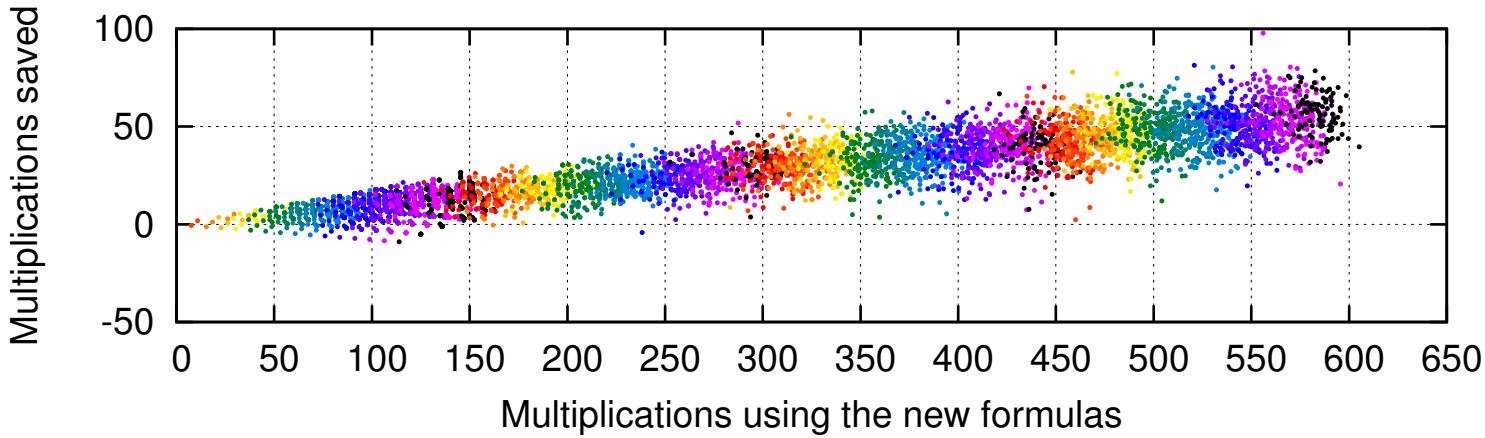
Our algorithm

We improved "tree-based" approach as follows:

- use not just $n-1$ and $n+1$, but all $n-c$
  where c is in a precomputed set S (including both $\pm$ values)
- add new nodes to n as follows:
  - **one** child node $n/2$ if n divisible by 2
  - **one** child node $n/3$ if n divisible by 3
  - **several** child nodes $n-c$, one for each $c \in S$
- continue searching until C chains are found
  rather than stopping with the first chain
  (use $C = 200$, choose the lowest-cost one)
- take lowest-weight B nodes at each level
  instead of the smallest B nodes
  define "weight" as cost $+ 8\log_2(n)$

Experiments and results

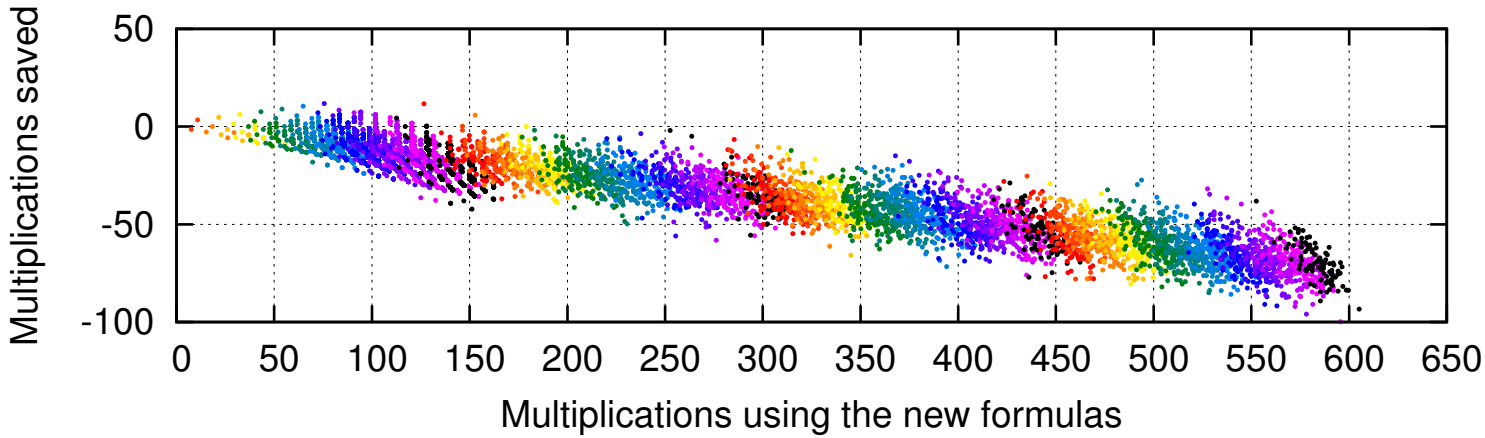
Twisted Hesssian VS short Weierstrass



- point (x, y) for 100 randomly sampled per b-bit integers n
  - b from 2 through 16, all b-bit integers
  - b from 17 through 64, randomly chosen b-bit 1000 integers per b
- $x\mathbf{M}$ are used to compute $P \to nP$ on twisted Hessian curves where $(x+y)\mathbf{M}$ are used for Weierstrass curves
- different colors represent different bit-size b

Twisted Hesssian VS twisted Edwards

Also considered double-and-add with signed sliding window of width 4



- point (x, y) for 100 randomly sampled per b-bit integers n
  - b from 2 through 16, all b-bit integers
  - b from 17 through 64, randomly chosen b-bit 1000 integers per b
- $x\mathbf{M}$ are used to compute $P \to nP$ on twisted Hessian curves where $(x+y)\mathbf{M}$ are used for Edwards curves
- different colors represent different bit-size b

Applications

Applications for which

(twisted) Hessian might be faster than Weierstrass

Applications

Applications for which
(twisted) Hessian might be faster than Weierstrass

- Secret scalar (concerns about constant-time)
  - "Exponentiating in Pairing Groups"
    by Bos, Costello and Naehrig

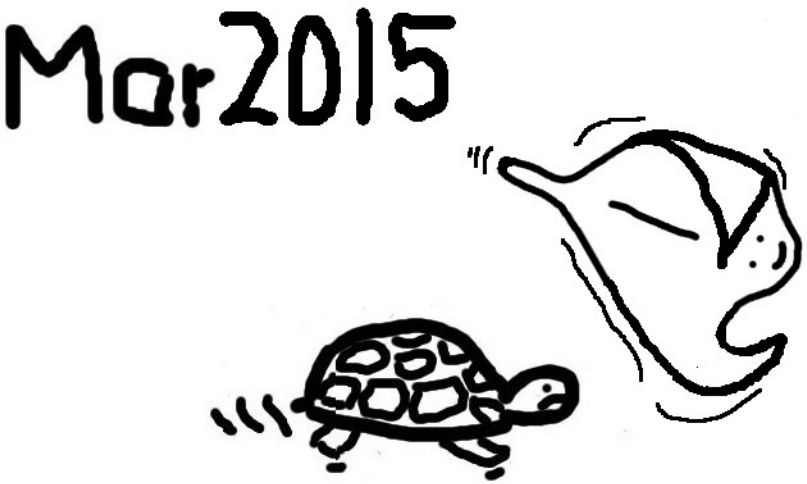
Applications

Applications for which

(twisted) Hessian might be faster than Weierstrass

- Secret scalar (concerns about constant-time)
  - "Exponentiating in Pairing Groups"
    by Bos, Costello and Naehrig
- Public scalar (do not worry about constant-time)
  - signature verifications
    e.g. $n_1P + n_2Q$
  - elliptic-curve computations
    e.g. ECM for integer factorization

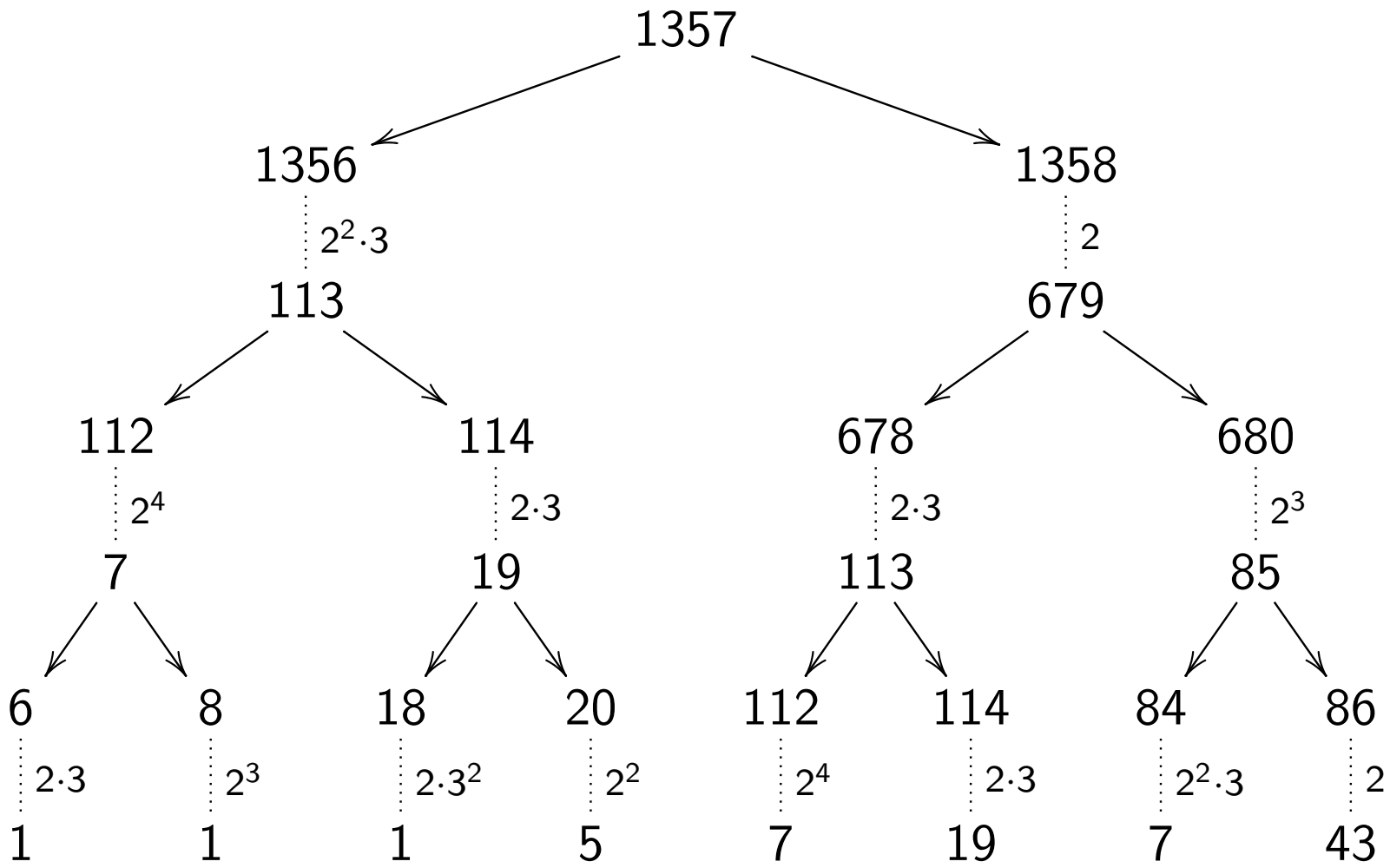

START


Mar2015

Optimal double-base chains

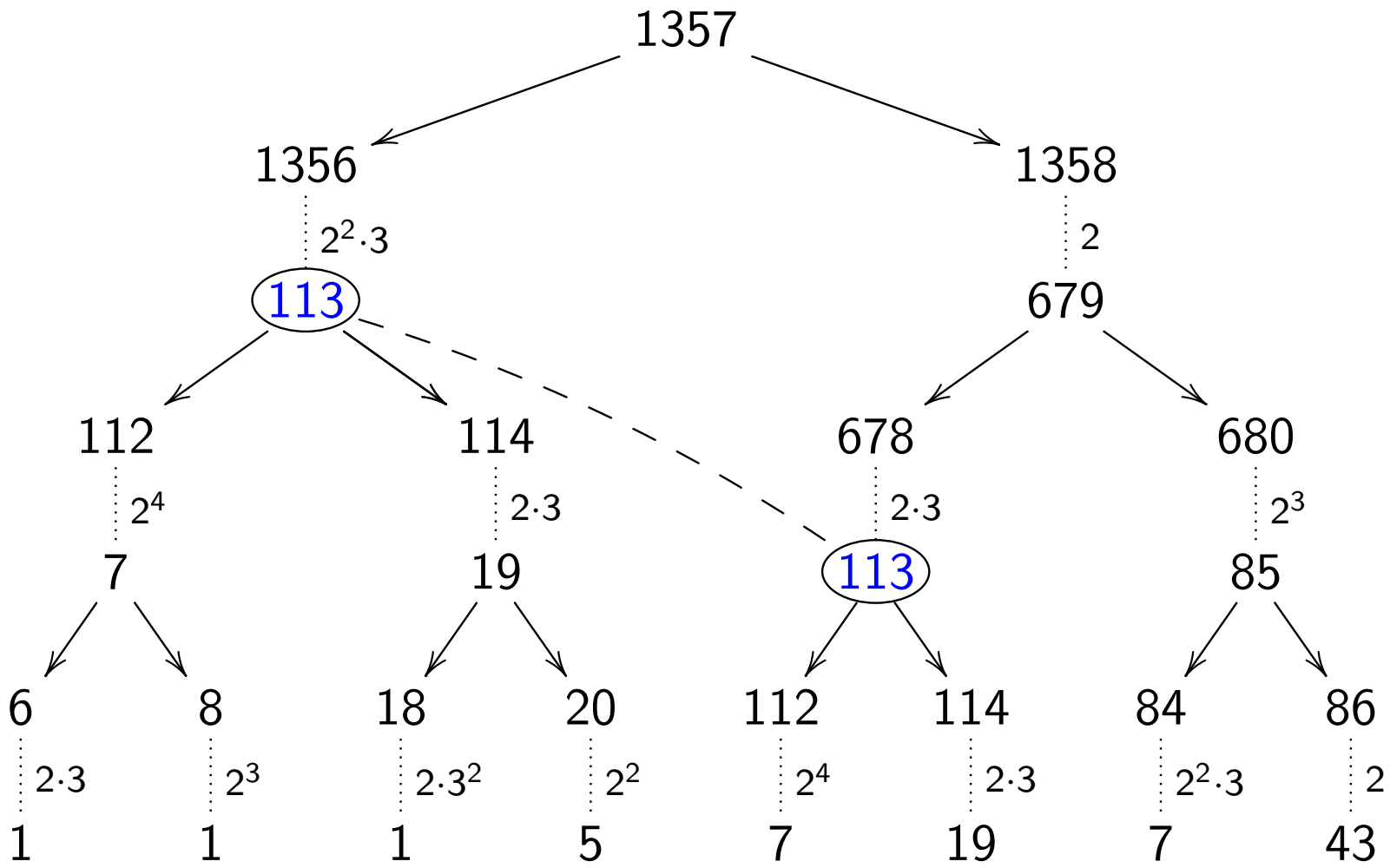
Tree search

(Doche and Habsieger 2008)



Tree search

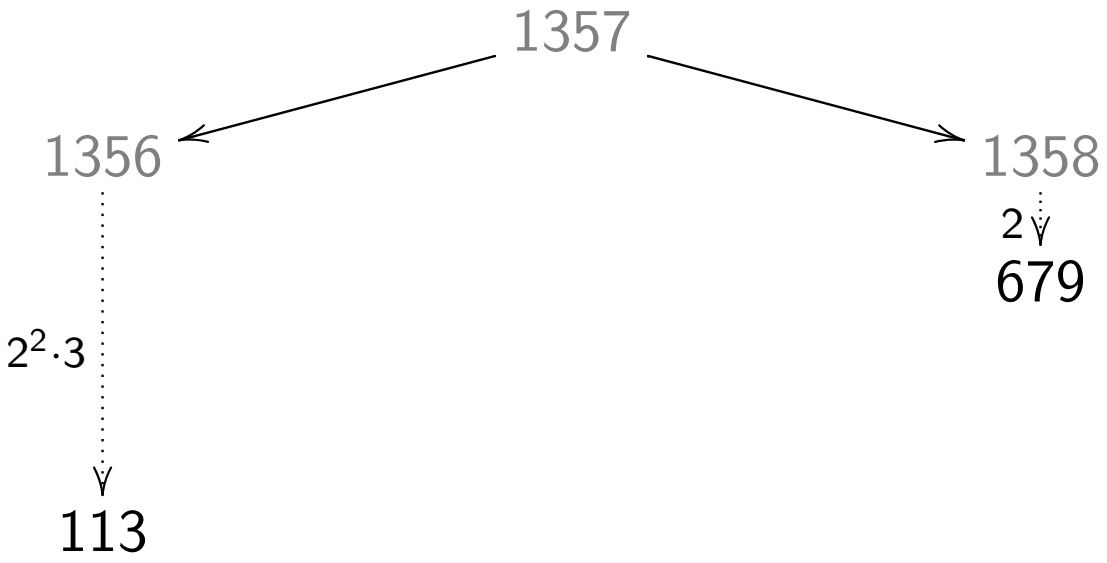
(Doche and Habsieger 2008)



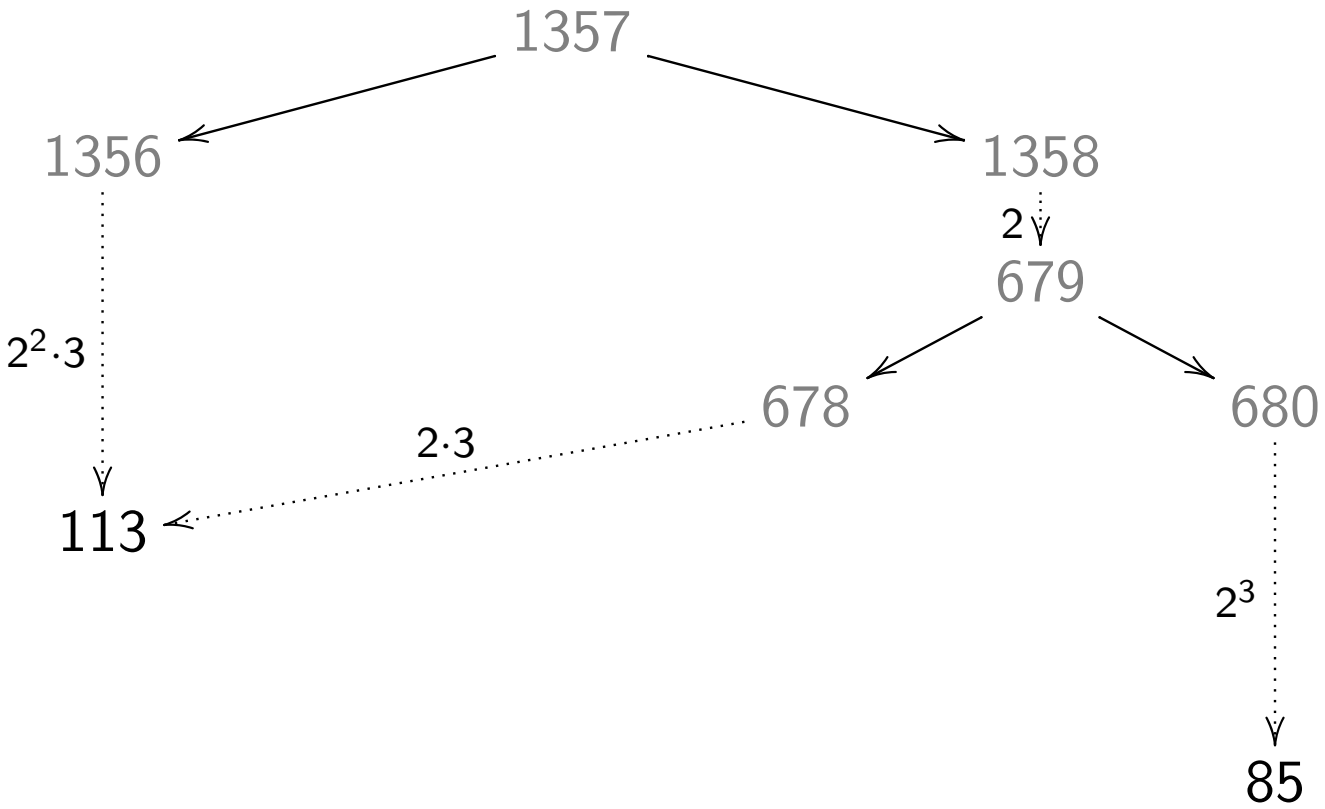
Directed-acyclic-graph search

1357

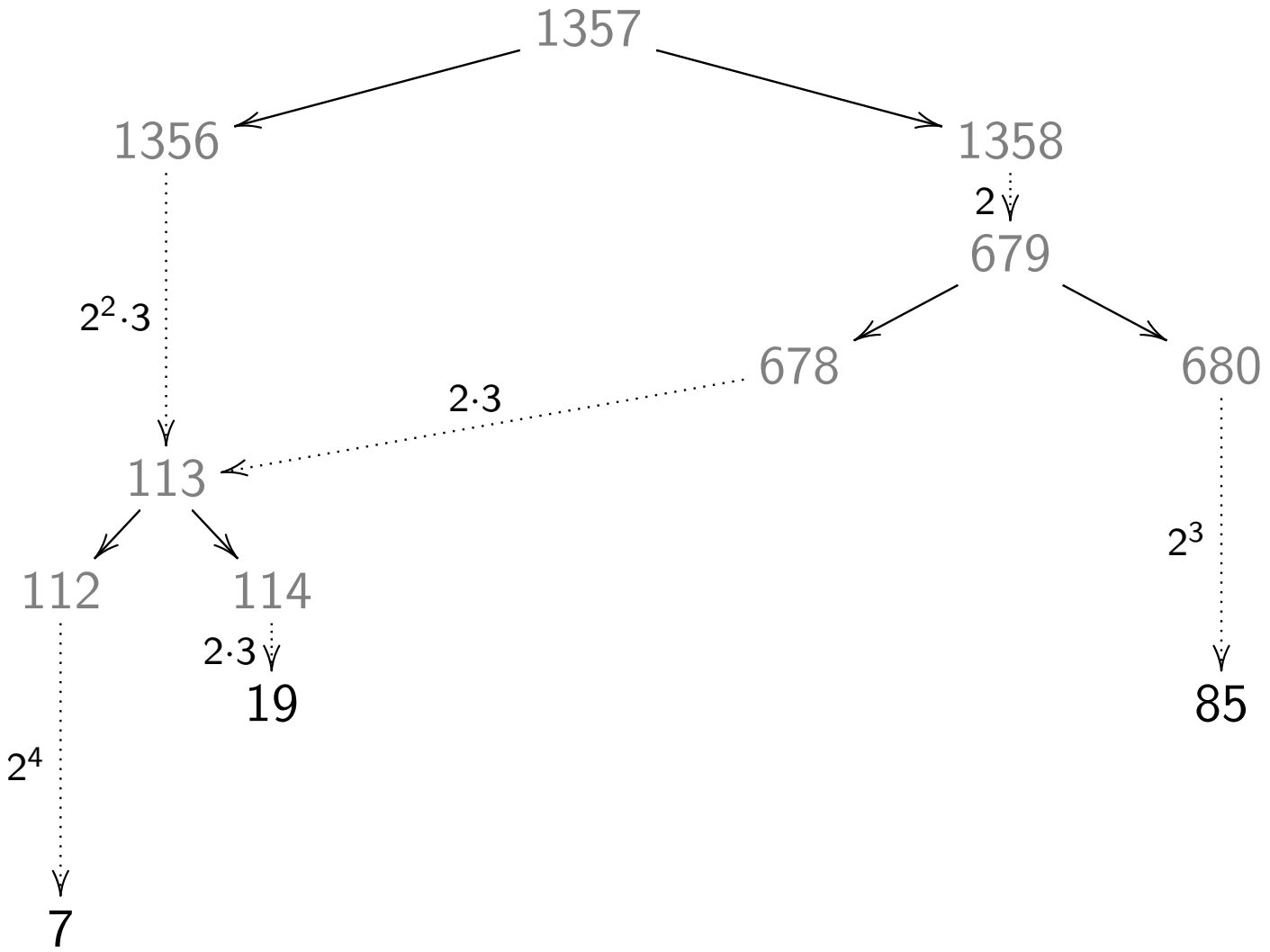
Directed-acyclic-graph search



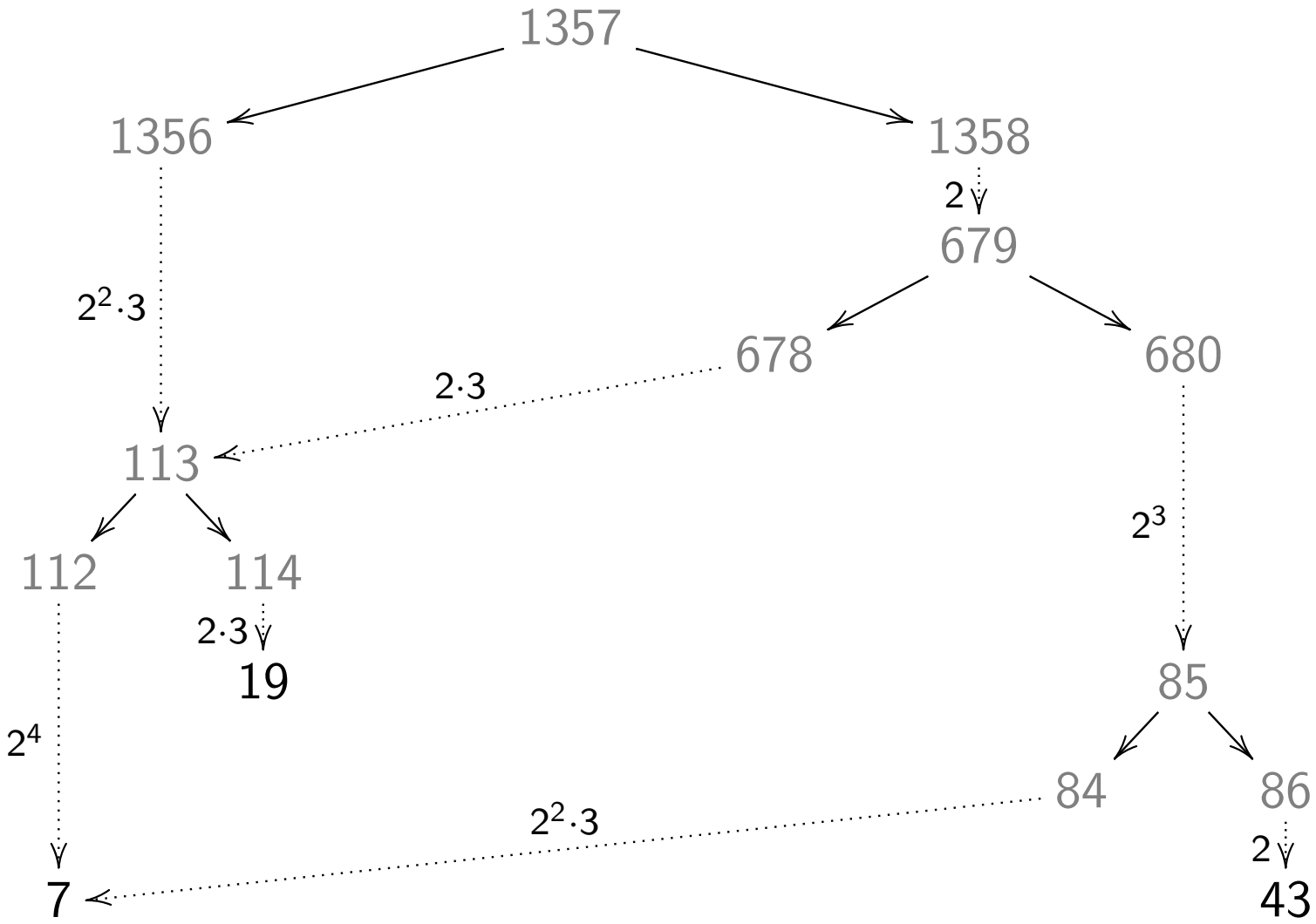
Directed-acyclic-graph search



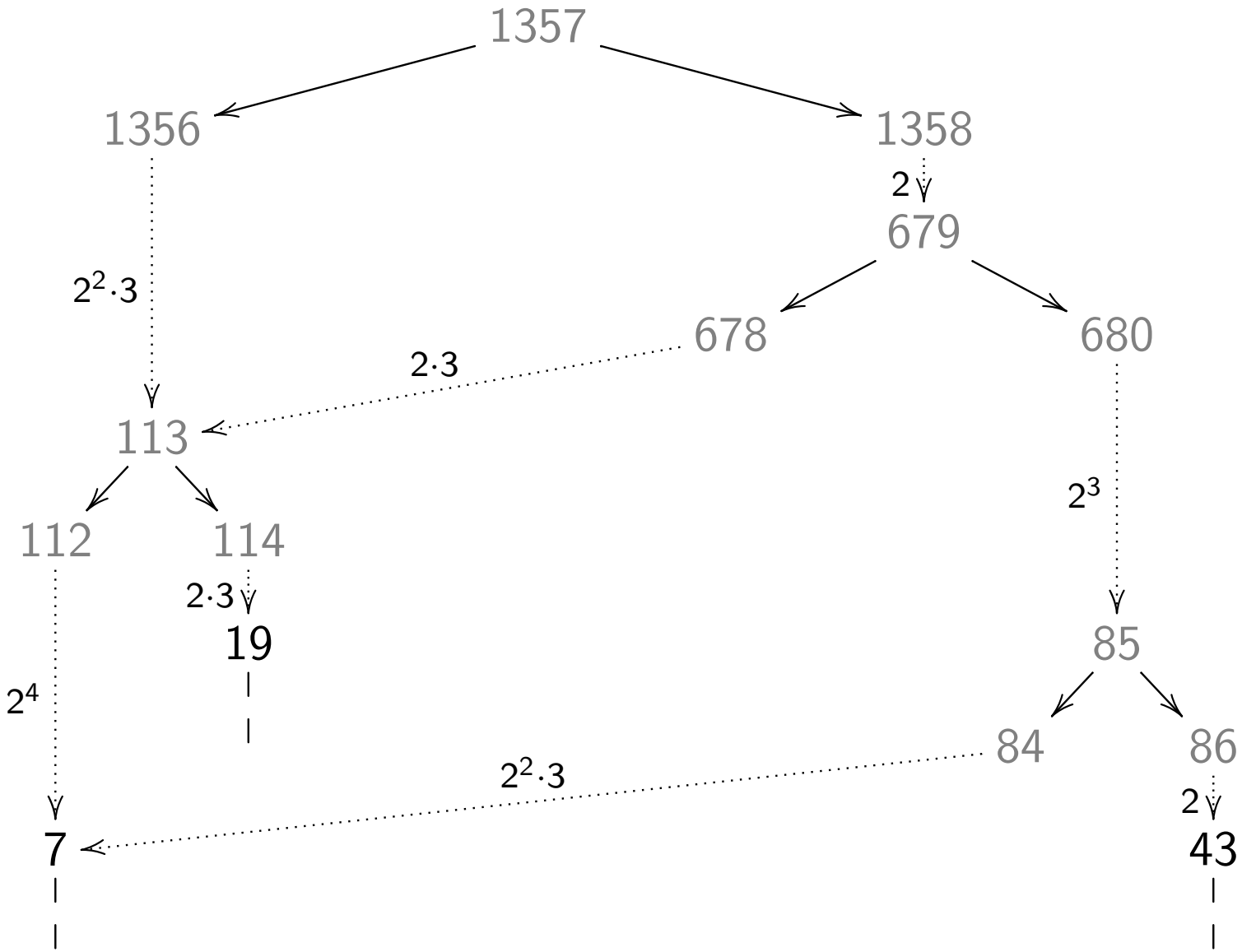
Directed-acyclic-graph search



Directed-acyclic-graph search



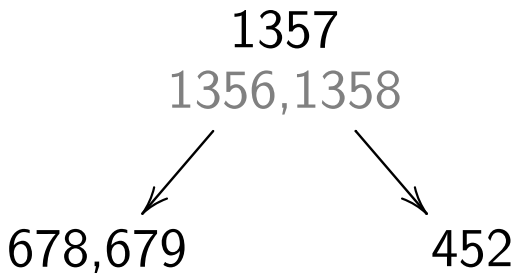
Directed-acyclic-graph search



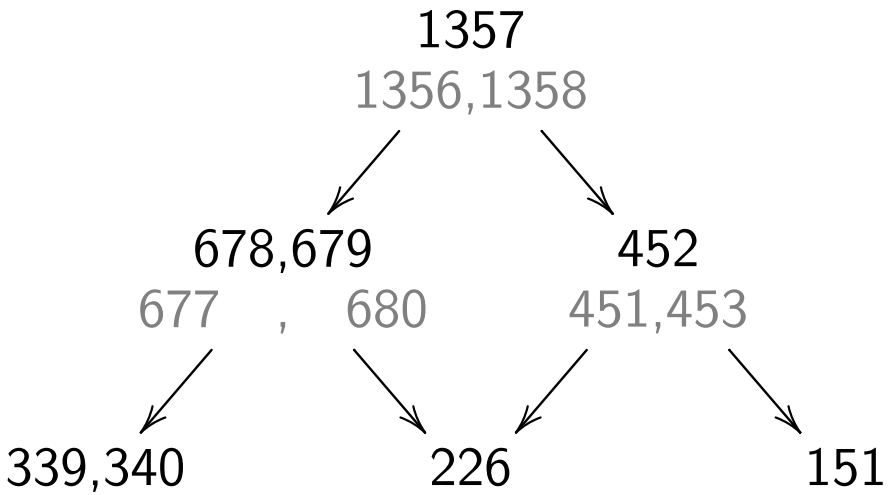
Rectangular DAG

1357

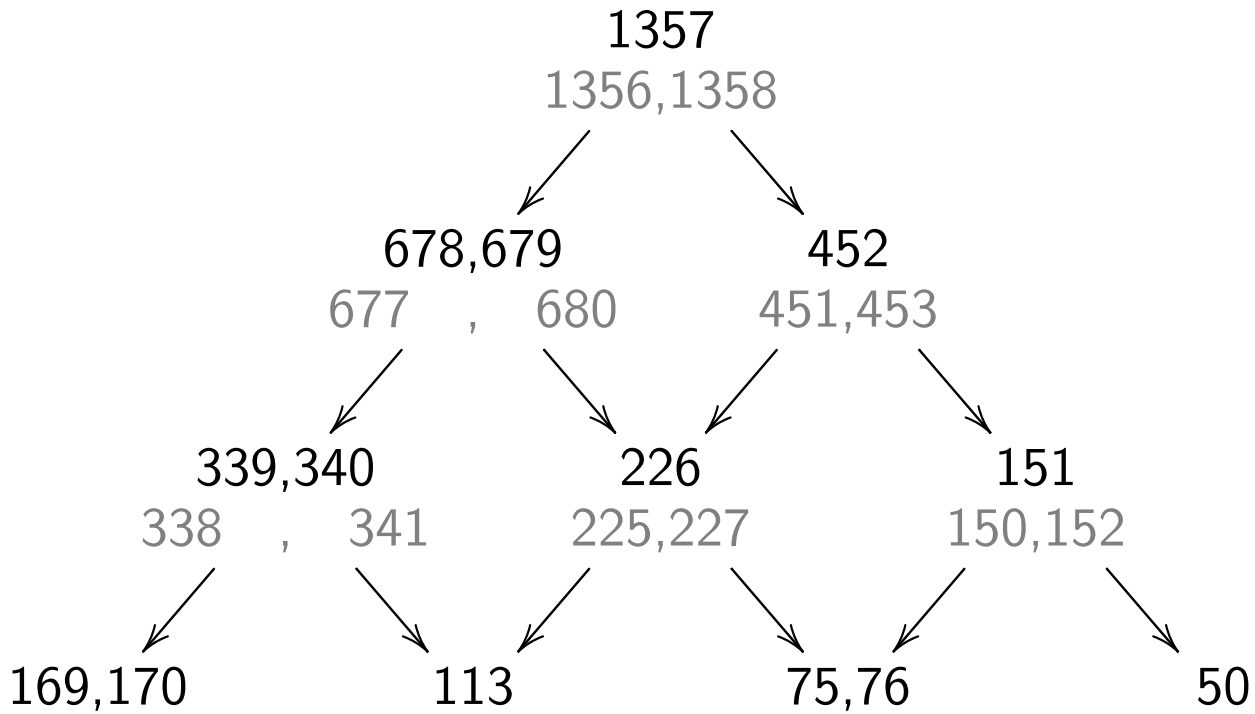
Rectangular DAG



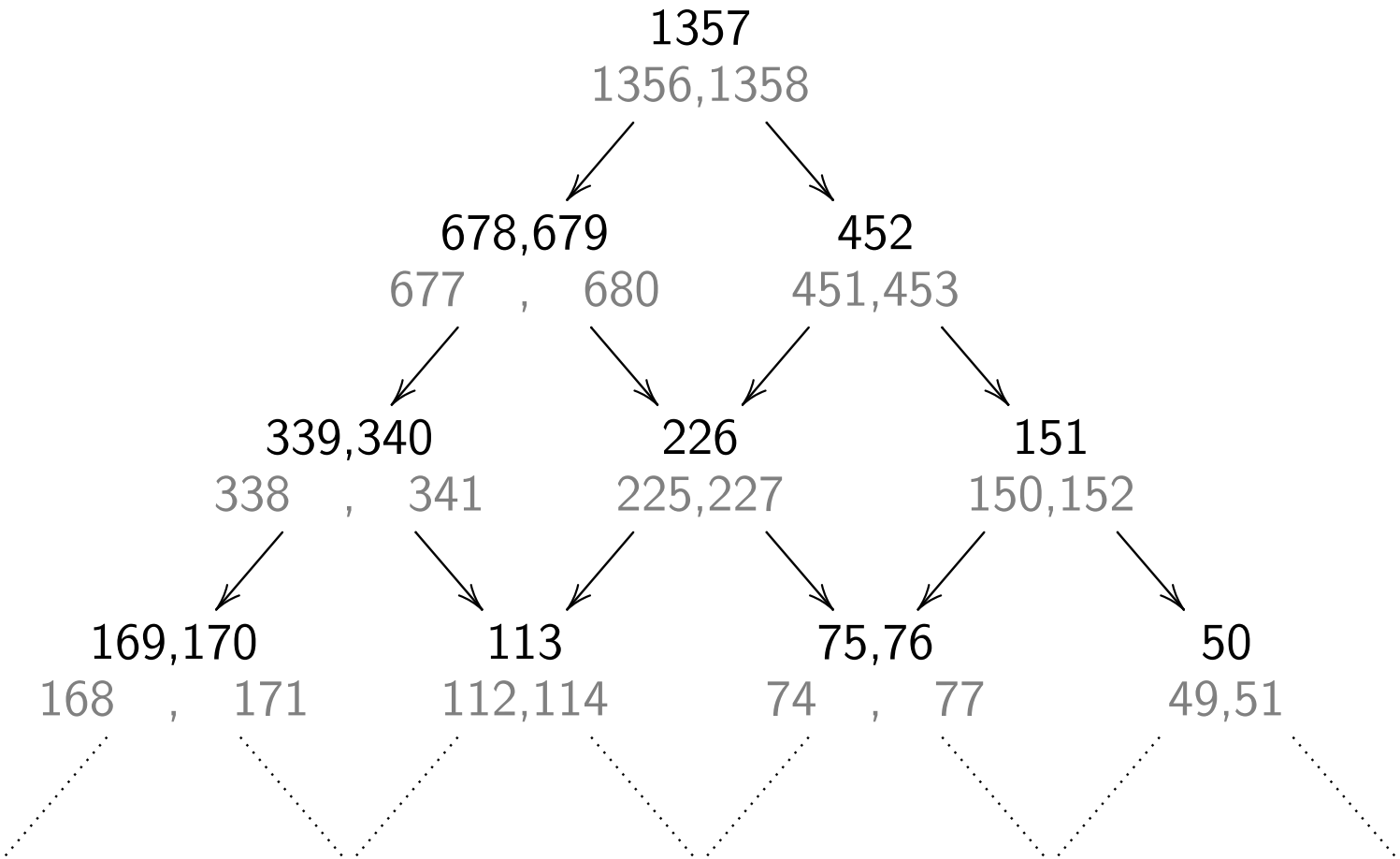
Rectangular DAG



Rectangular DAG



Rectangular DAG



Extra additions

- Why not always completely factor out 2 and 3?
- Is it useful to consider additions every step?

Extra additions

- Why not always completely factor out 2 and 3?
- Is it useful to consider additions every step?

- Yes!!!

  e.g. $n = 28$

  (assume TPL $= 10.8\mathbf{M}$, DBL $= 7.6\mathbf{M}$, ADD $= 11\mathbf{M}$)

  $28 = 2^2(2 \cdot 3 + 1)$ cost $= 44.6$

  $28 = 3^3 + 1$ cost $= 43.4$

Extra additions

- Why not always completely factor out 2 and 3?
- Is it useful to consider additions every step?

- Yes!!!

  e.g. $n = 28$

  (assume TPL $= 10.8\mathbf{M}$, DBL $= 7.6\mathbf{M}$, ADD $= 11\mathbf{M}$)

  $28 = 2^2(2 \cdot 3 + 1)$ cost $= 44.6$

  $28 = 3^3 + 1$ cost $= 43.4$

- Is this method worth it?

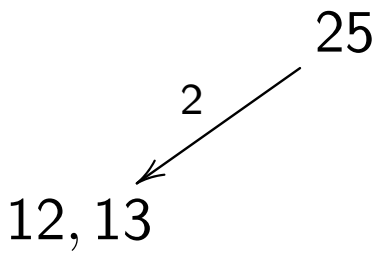
Residue classes

- Faster computation of the chain working with smaller number
  e.g. $1357 \equiv 25 \bmod 2^2 \cdot 3^2$ (11 bits $\rightarrow$ 5 bits)

$$25$$

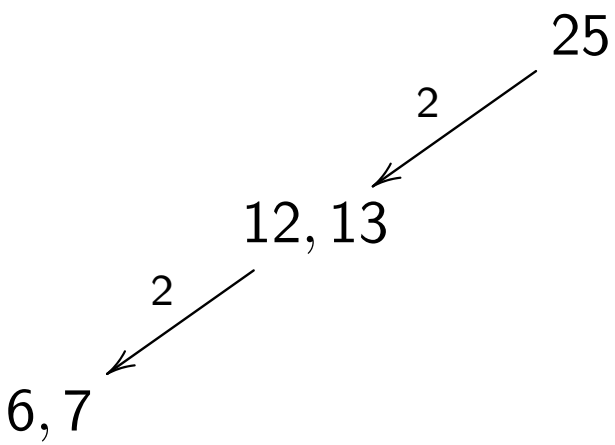
Residue classes

- Faster computation of the chain working with smaller number
  e.g. $1357 \equiv 25 \bmod 2^2 \cdot 3^2$ (11 bits $\rightarrow$ 5 bits)



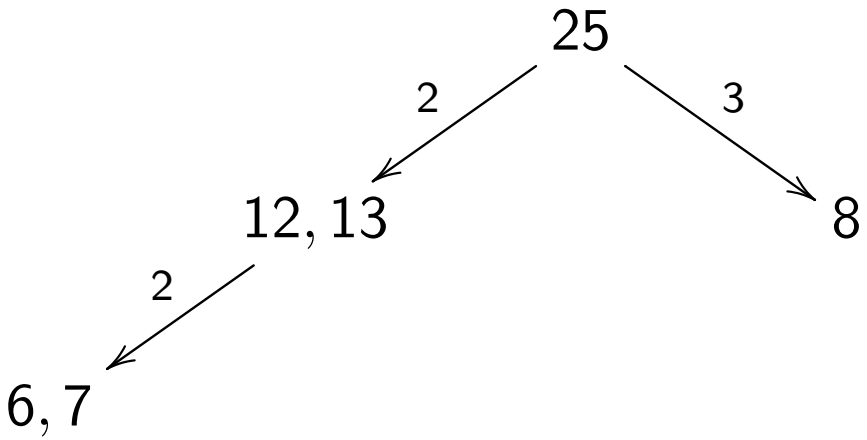
Residue classes

- Faster computation of the chain working with smaller number
  e.g. $1357 \equiv 25 \bmod 2^2 \cdot 3^2$ (11 bits $\rightarrow$ 5 bits)



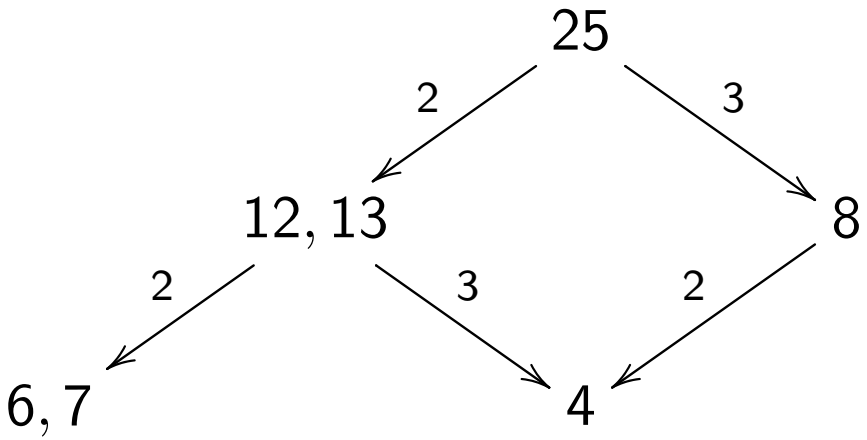
Residue classes

- Faster computation of the chain working with smaller number
  e.g. $1357 \equiv 25 \bmod 2^2 \cdot 3^2$ (11 bits $\rightarrow$ 5 bits)



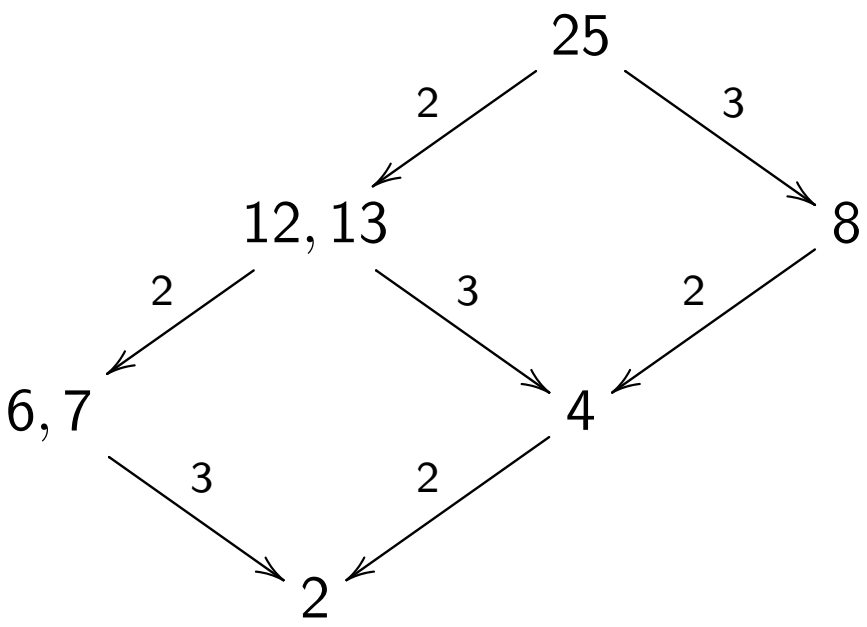
Residue classes

- Faster computation of the chain working with smaller number
  e.g. $1357 \equiv 25 \bmod 2^2 \cdot 3^2$ (11 bits $\rightarrow$ 5 bits)



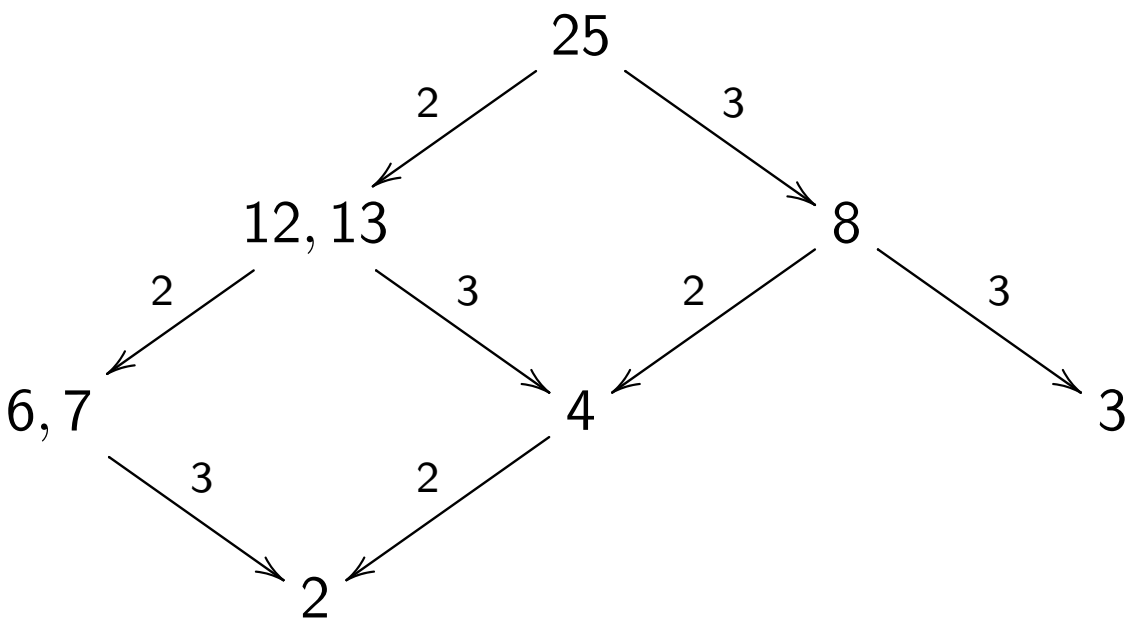
Residue classes

- Faster computation of the chain working with smaller number
  e.g. $1357 \equiv 25 \bmod 2^2 \cdot 3^2$ (11 bits $\rightarrow$ 5 bits)



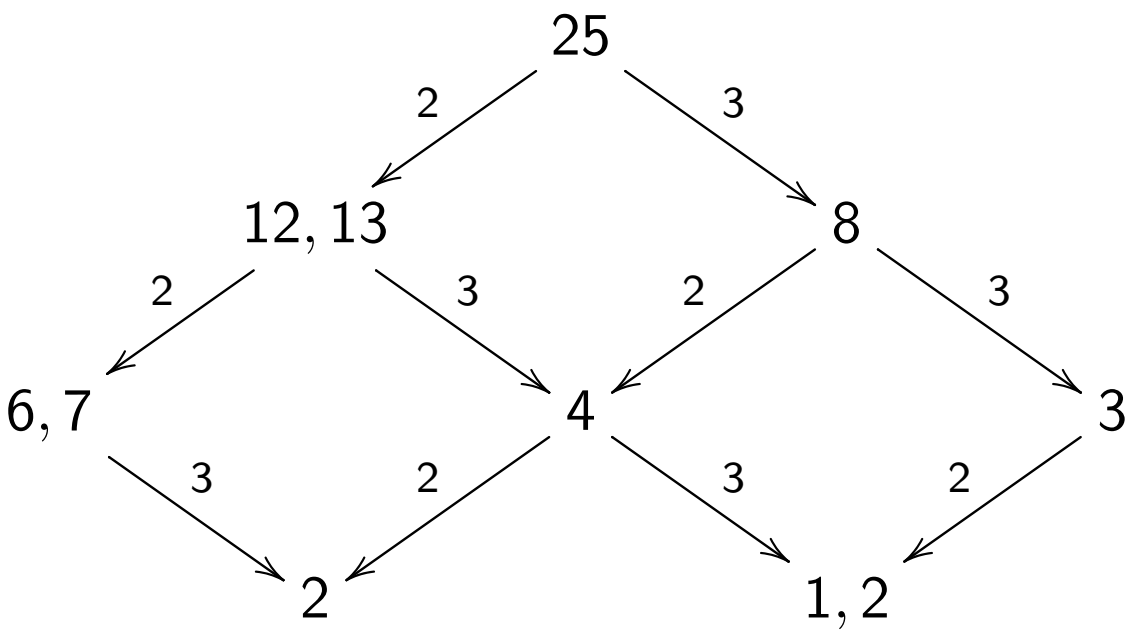
Residue classes

- Faster computation of the chain working with smaller number
  e.g. $1357 \equiv 25 \bmod 2^2 \cdot 3^2$ (11 bits $\rightarrow$ 5 bits)



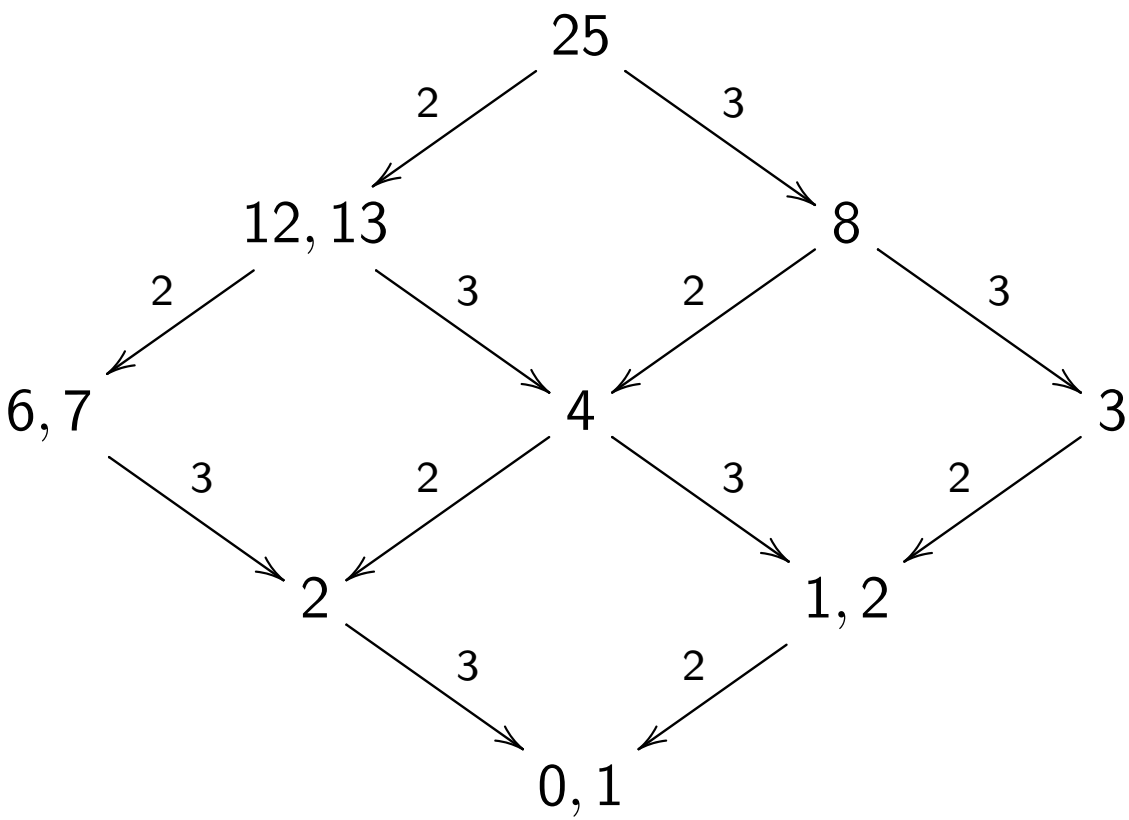
Residue classes

- Faster computation of the chain working with smaller number
  e.g. $1357 \equiv 25 \bmod 2^2 \cdot 3^2$ (11 bits $\rightarrow$ 5 bits)

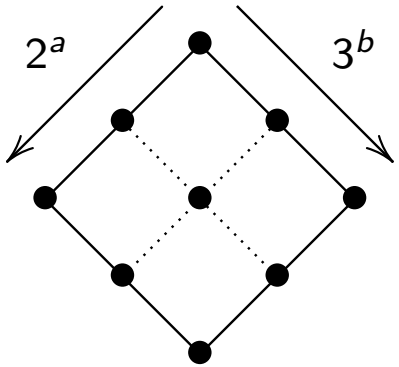


Residue classes

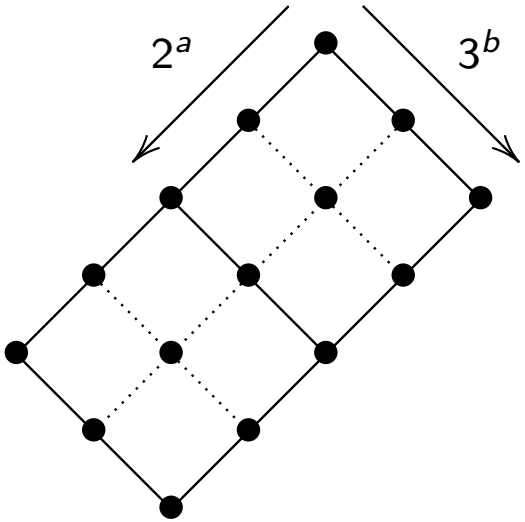
- Faster computation of the chain working with smaller number
  e.g. $1357 \equiv 25 \bmod 2^2 \cdot 3^2$ (11 bits $\rightarrow$ 5 bits)



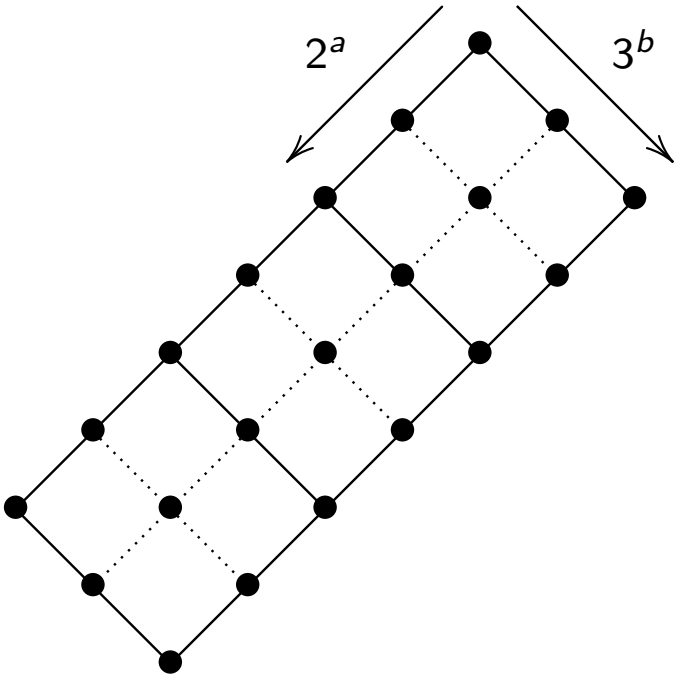
Residue classes



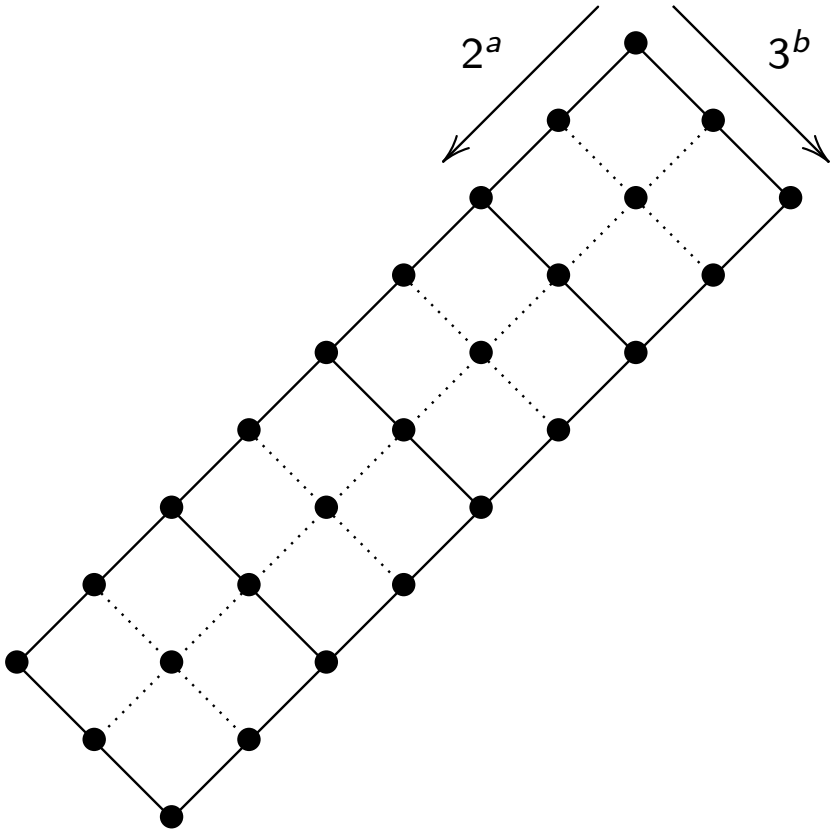
Residue classes



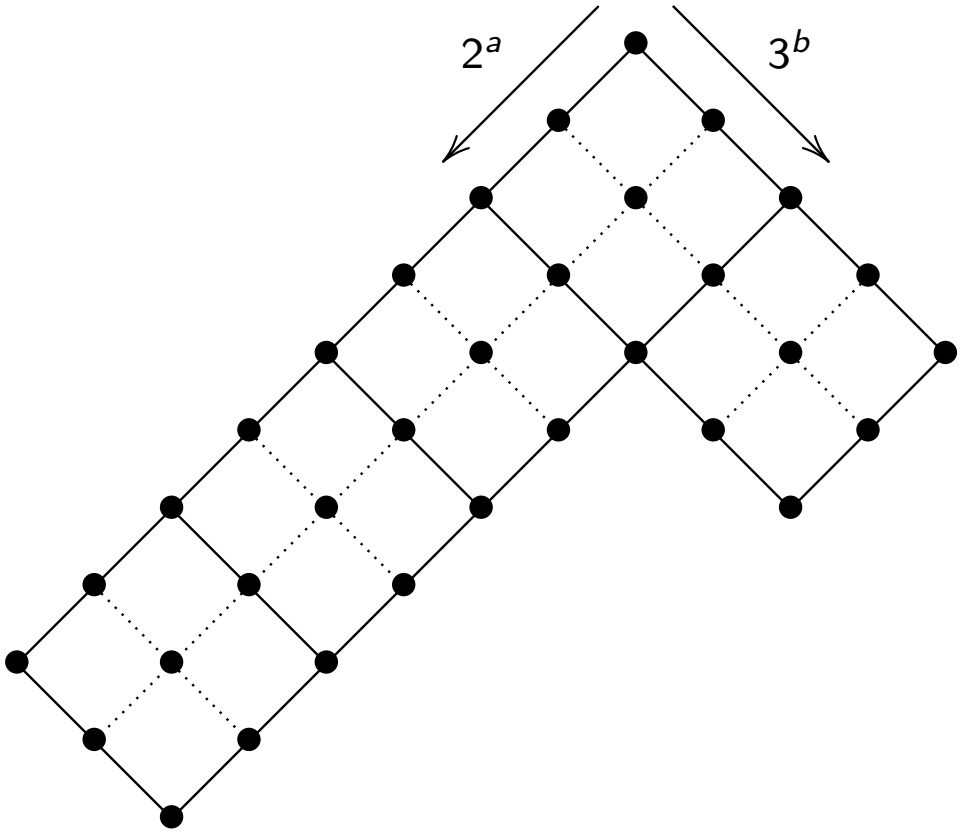
Residue classes



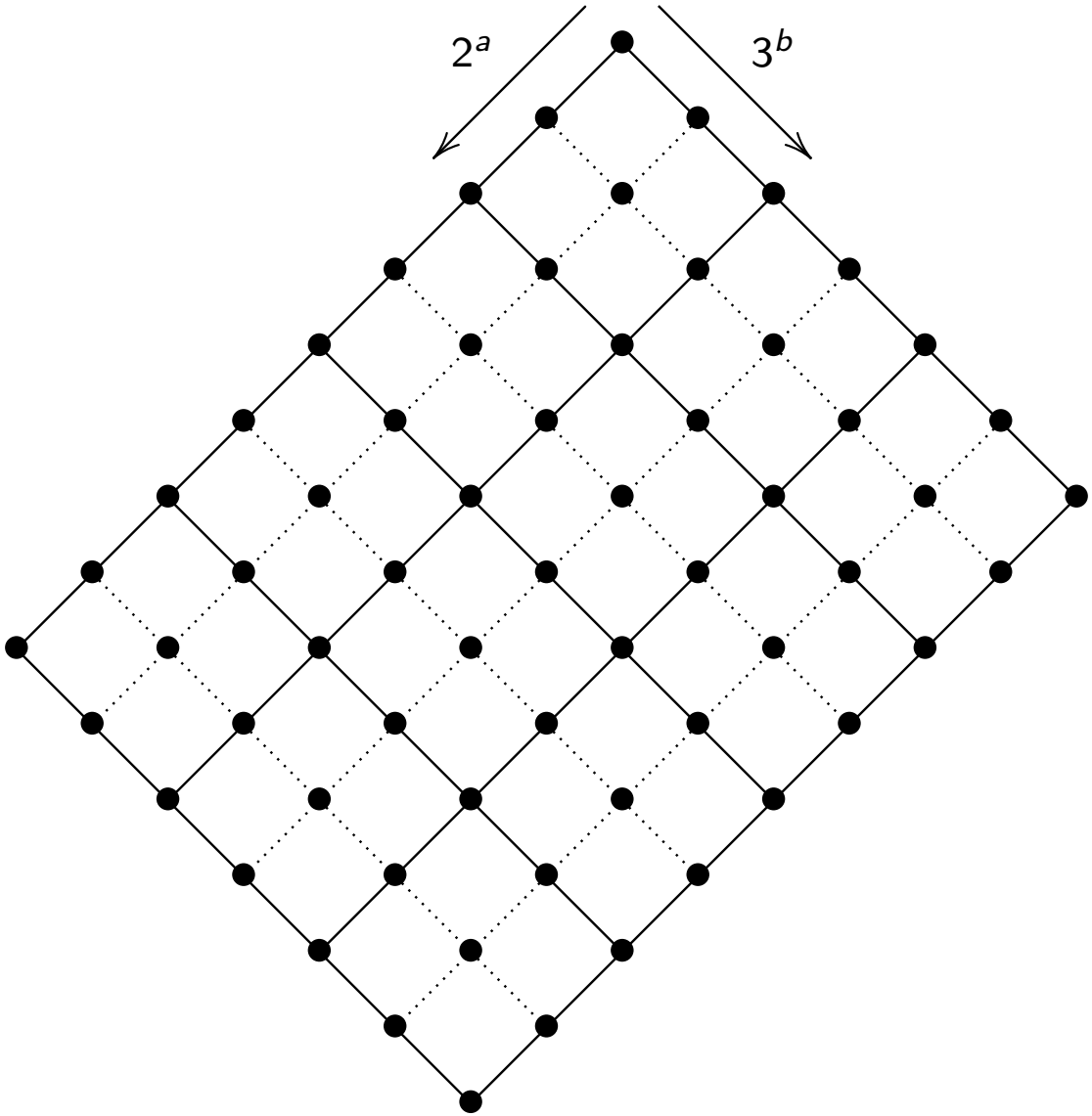
Residue classes



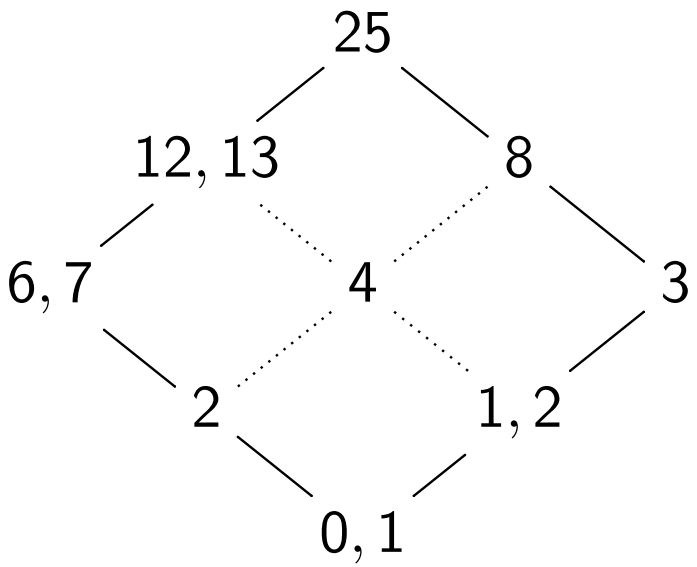
Residue classes



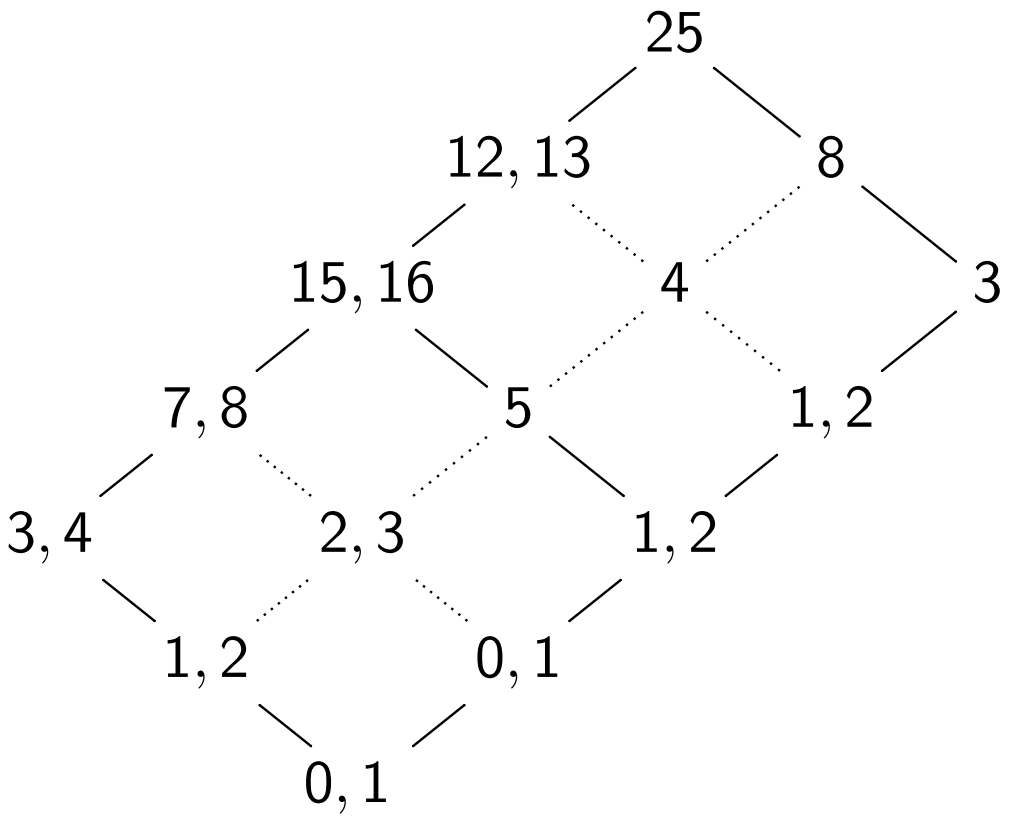
Residue classes



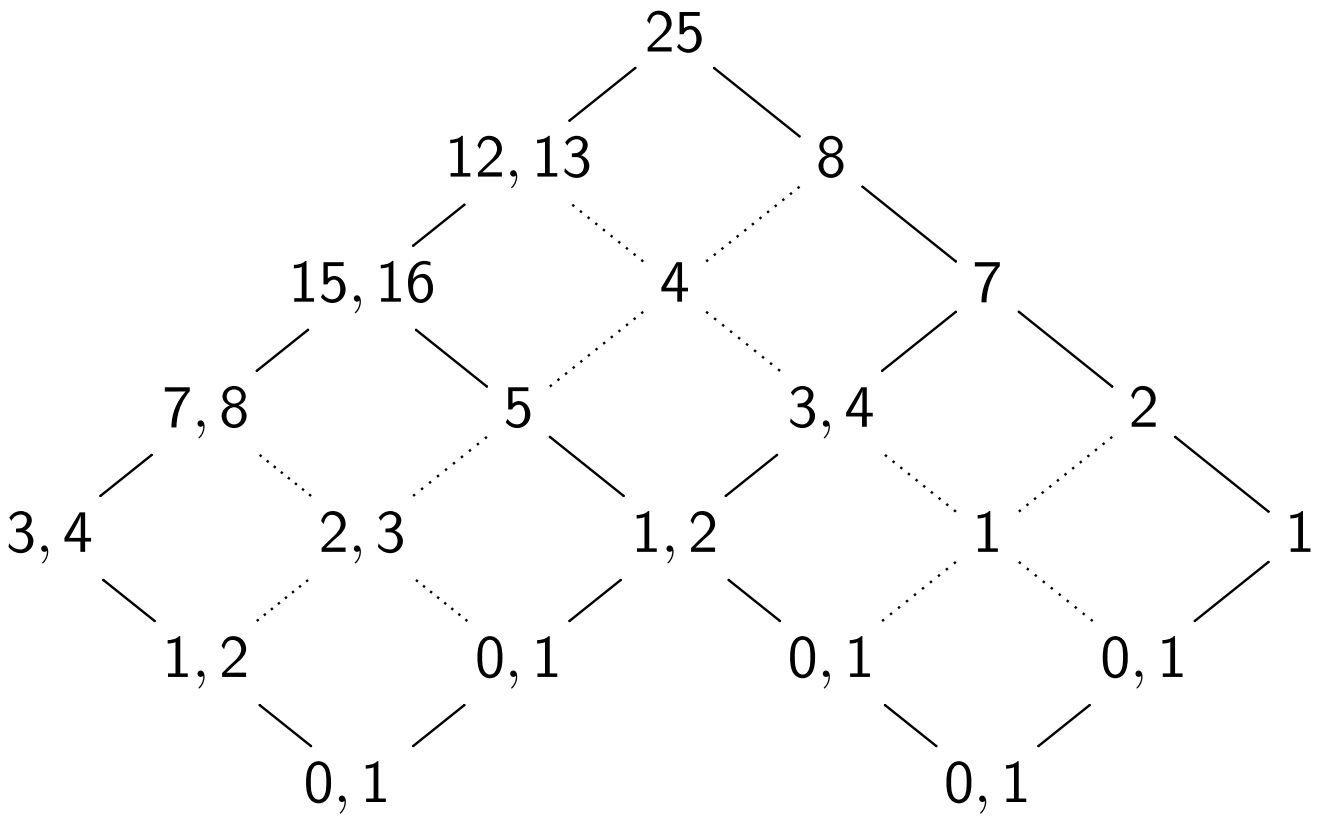
Residue classes



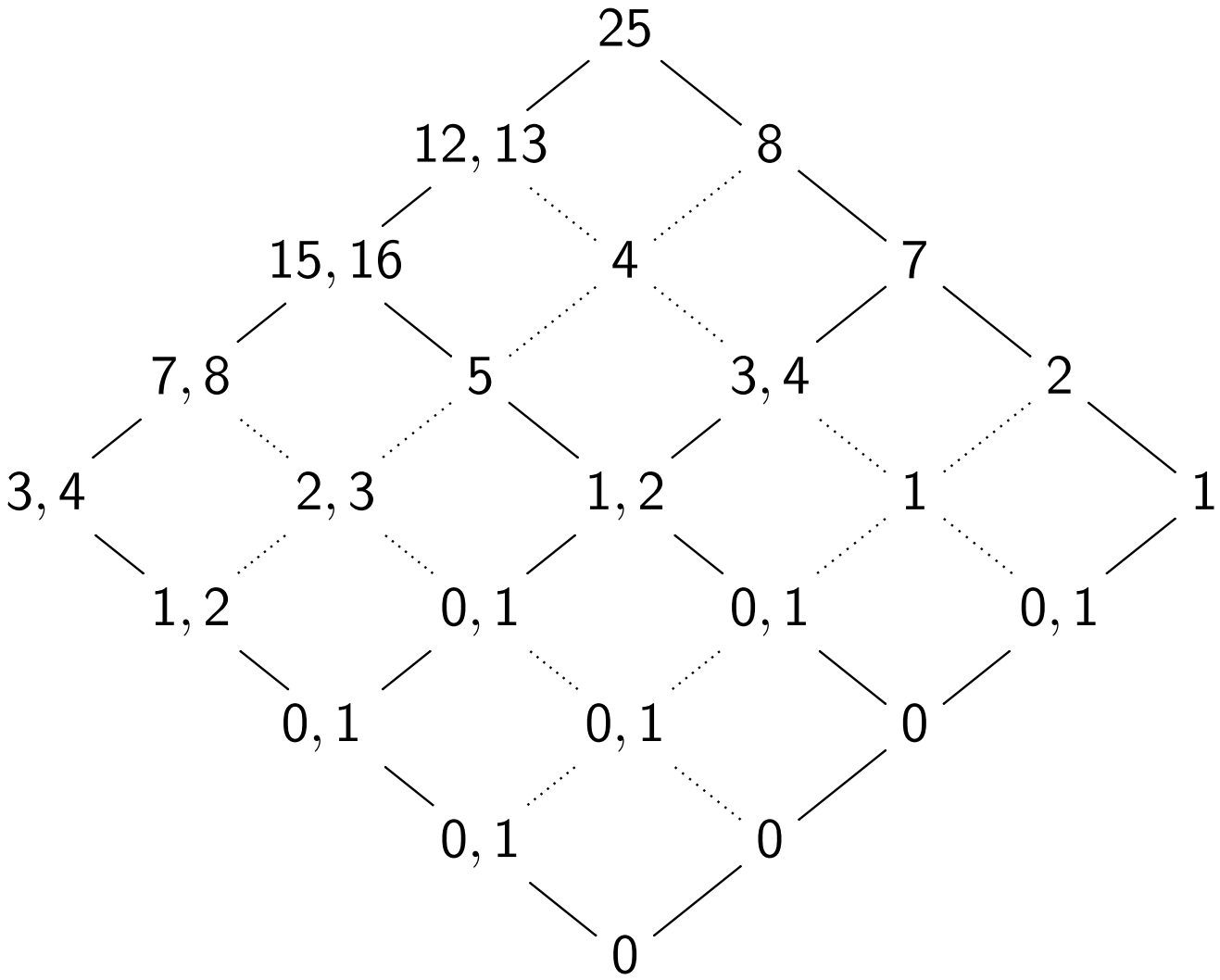
Residue classes



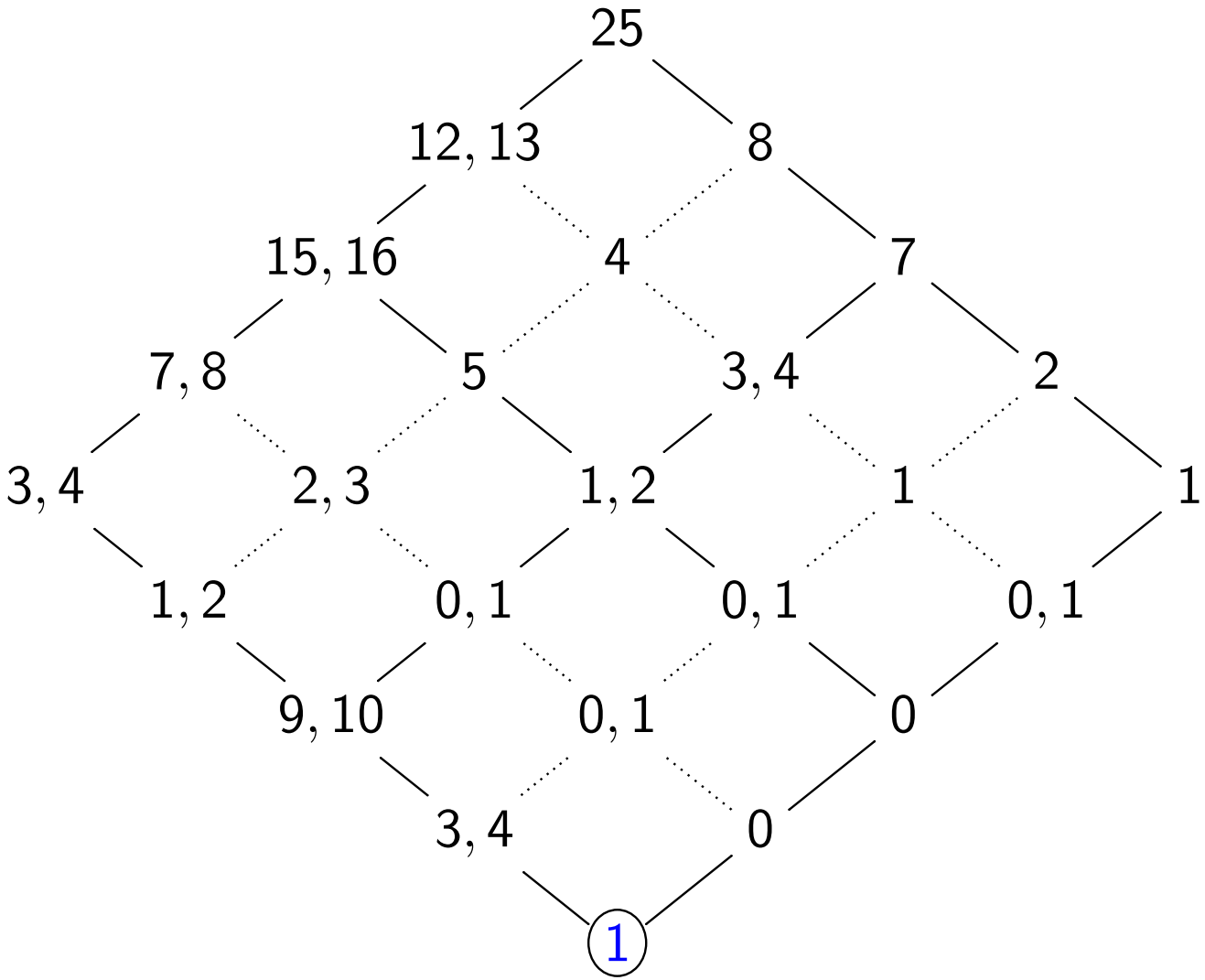
Residue classes



Residue classes



Residue classes



Results

Cost per bit to compute scalar multiplication for 256-bit

Curve shape	**S**/**M** ratio		
	1	0.8	0.67
Jacobian-3	10.20950	9.12516	8.39722
Twisted Hessian (new formulas)	9.16351	8.52279	8.09017
Twisted Edwards	8.27195	7.52247	7.01979
Twisted Edwards (new formulas)	8.20036	7.47415	6.97923

Results

Cost per bit to compute scalar multiplication for 256-bit

Curve shape	**S**/**M** ratio		
	1	0.8	0.67
Jacobian-3	10.20950	9.12516	8.39722
Twisted Hessian (new formulas)	9.16351	8.52279	8.09017
Twisted Edwards	8.27195	7.52247	7.01979
Twisted Edwards (new formulas)	8.20036	7.47415	6.97923

Cost of 256-bit scalar multiplication using new formulas twisted Edwards

Base	Mults	Mults/ℓ	S
double	2092.60 [Doche]	8.17422	$\pm\{0,1\}$
double	1994.84 (new)	7.79233	$\pm\{0,1\}$
single	1950.60 [Hisil]	7.61953	$\pm\{0,1,3,5,7,9,11,13,15\}$
single	1938.57 (new)	7.57252	$\pm\{0,1,3,5,7,\ldots,21\}$
double	1913.14 (new)	7.47320	$\pm\{0,1,5,7,11,13,17,19\}$
double	1912.91 (new)	7.47229	$\pm\{0,1,2,4,5,7,11,13,17,19\}$

Results

Cost to compute double-base double-scalar multiplication on twisted Edwards

B	Method	$\lvert S\rvert$	192	256	320	384	448	512
double	Tree-JBT [Doche]	4	1953	2602	3248	3896	4545	5197
	Tree-JBT$_5$ [Doche]	6	1920	2543	3168	3792	4414	5042
	Tree-JBT$_7$ [Doche]	8	1907	2521	3137	3753	4365	4980
	Tree-JBT$_{5^2}$ [Doche]	12	1890	2485	3079	3677	4270	4862
single	sliding (ω=3) (new)	4	1884	2494	3103	3715	4324	4935
	sliding (ω=4) (new)	8	1838	2424	3009	3595	4181	4767
	sliding (ω=5) (new)	16	1836	2391	2944	3500	4055	4610
double	rrDAG (new)	4	1768	2352	2937	3521	4105	4690
	rrDAG$_5$ (new)	6	1748	2315	2883	3450	4018	4585
	rrDAG$_7$ (new)	8	1734	2292	2851	3410	3969	4527
	rrDAG$_{5^2}$ (new)	12	1709	2252	2794	3337	3879	4422