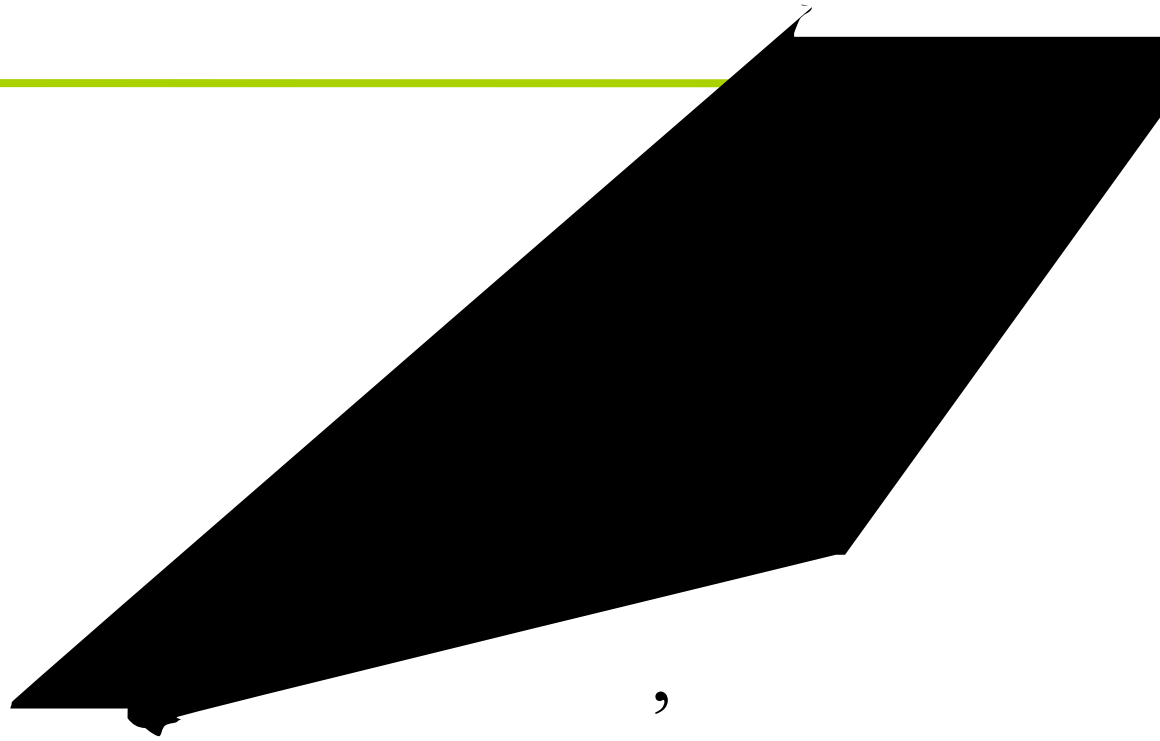
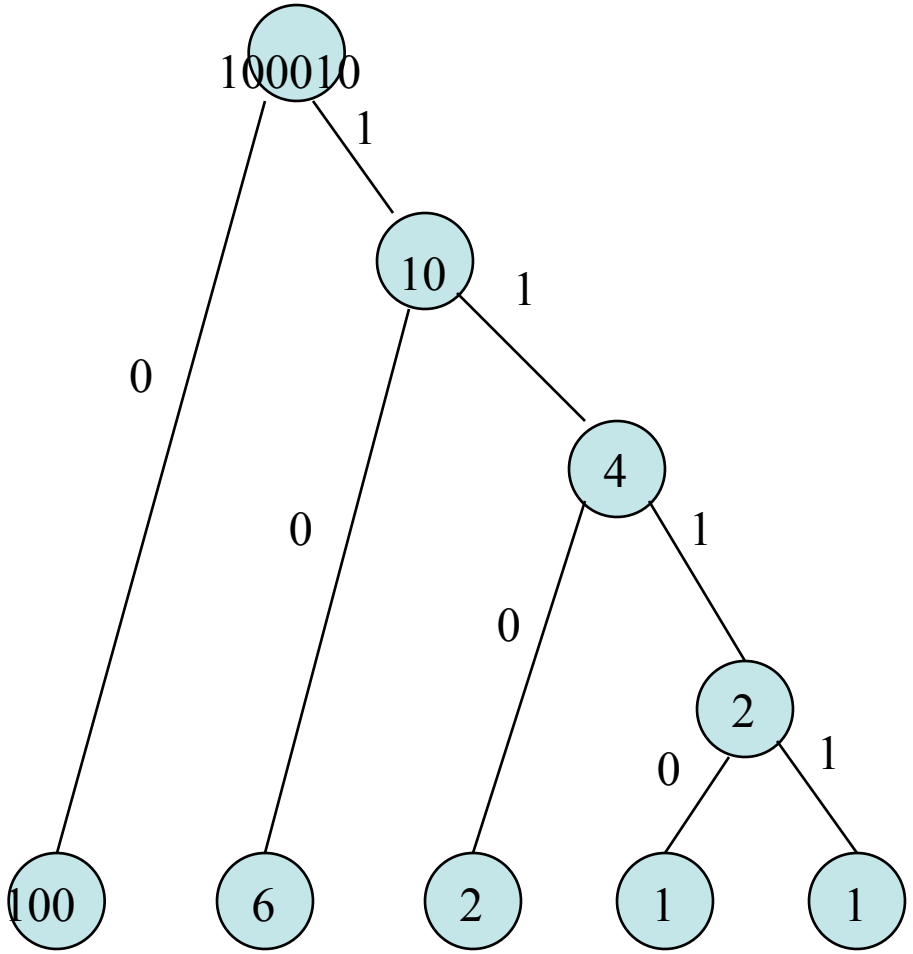
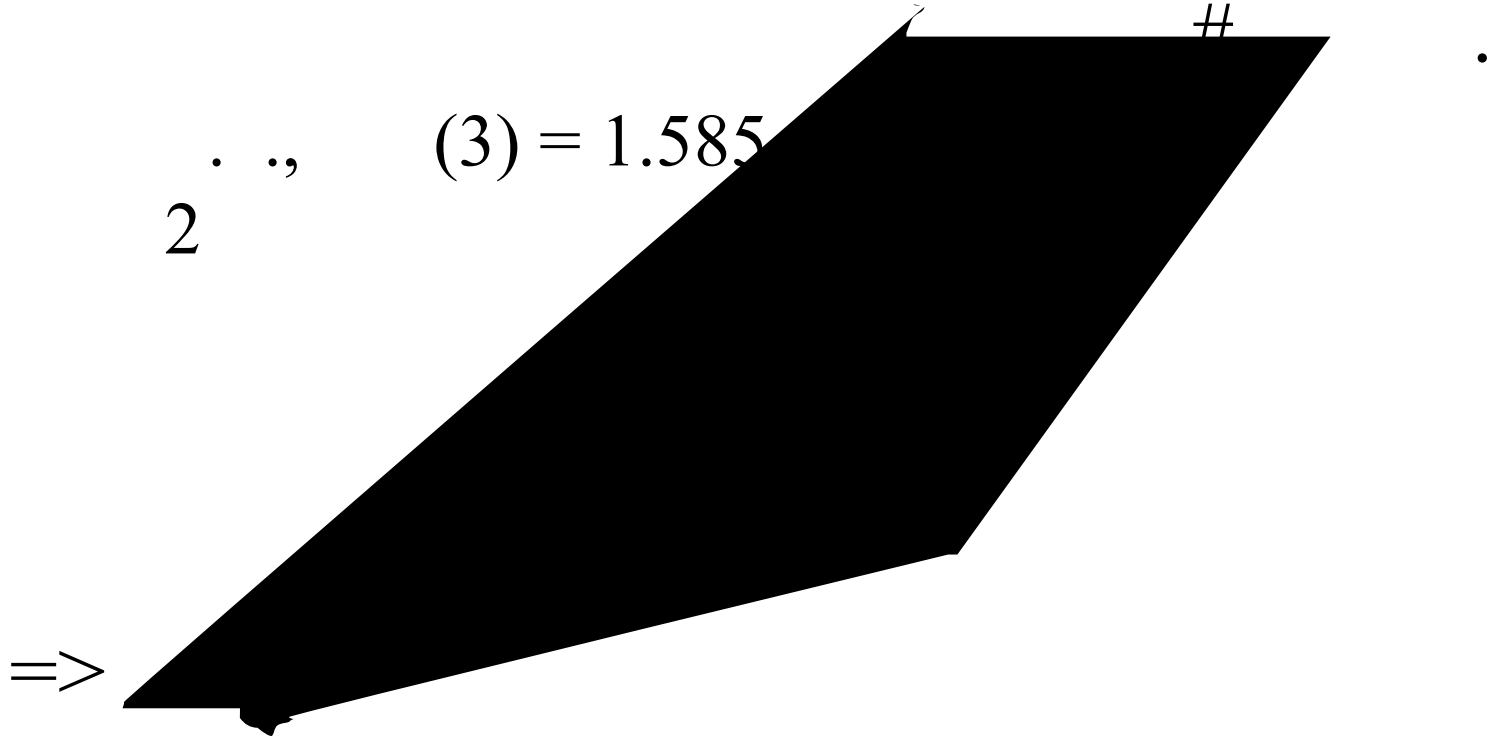
2:  ,

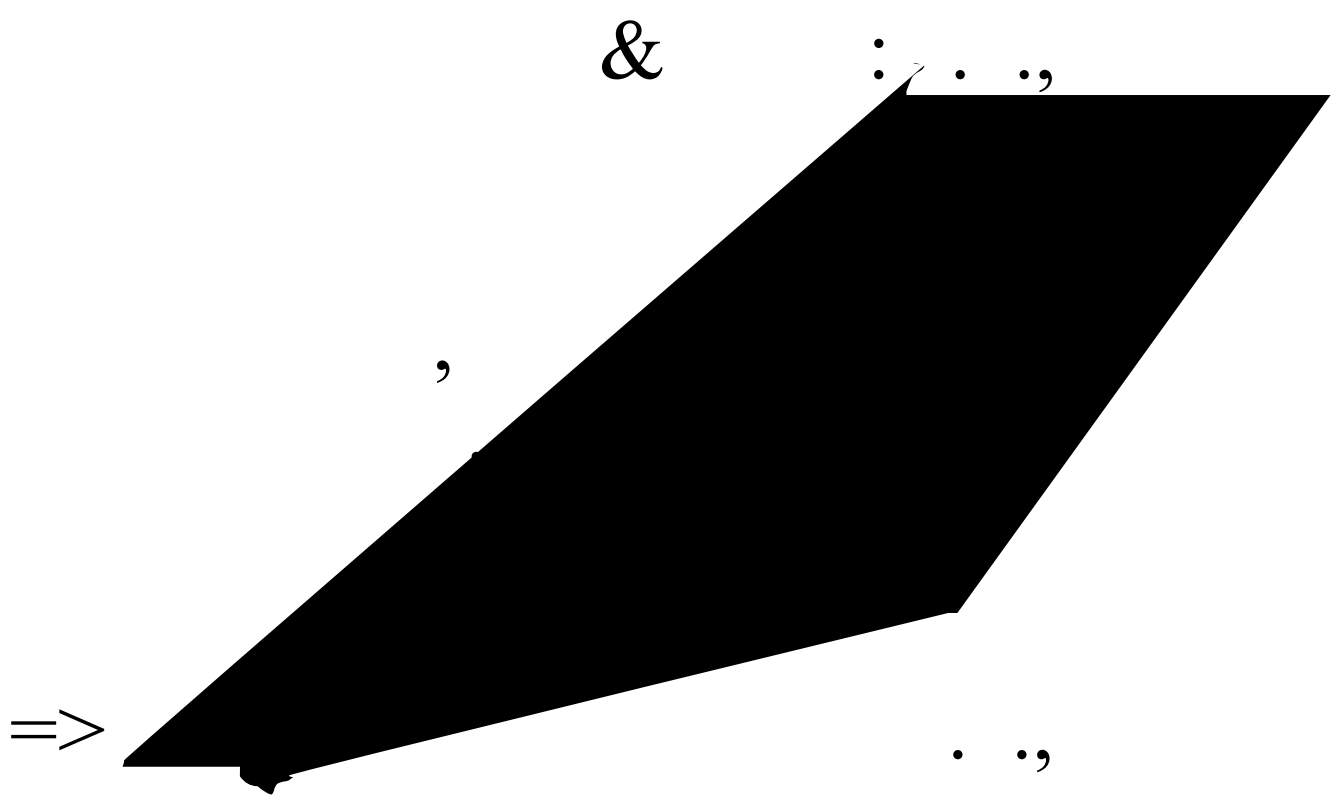
	100000	0
	6	10
	2	110
	1	1110
	1	1111

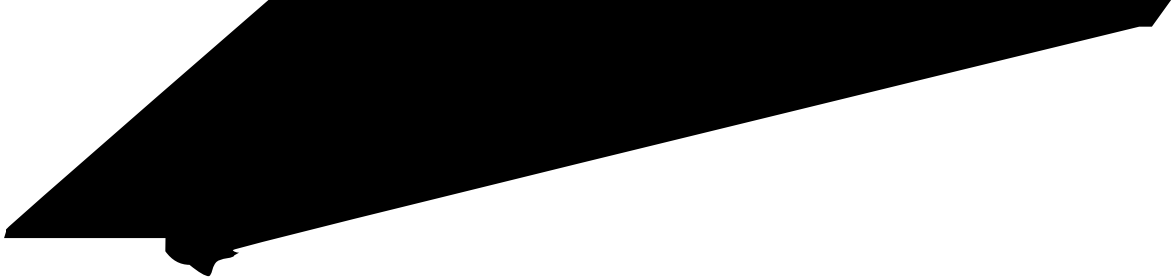


$= 0.9999 \quad 1.0001 + 0.00006 \quad 16668.333$
$+ \quad + 1/100010 \quad 100010$
$\approx \mathbf{0.00}$

$= (100000*1 + \quad)/100010$
$\approx \mathbf{1}$







Encoder

(

)

Decoder

(

)

&

.

( )

Encoder

(

)

Decoder

(

)

( )

Encoder

(

)

Decoder

(

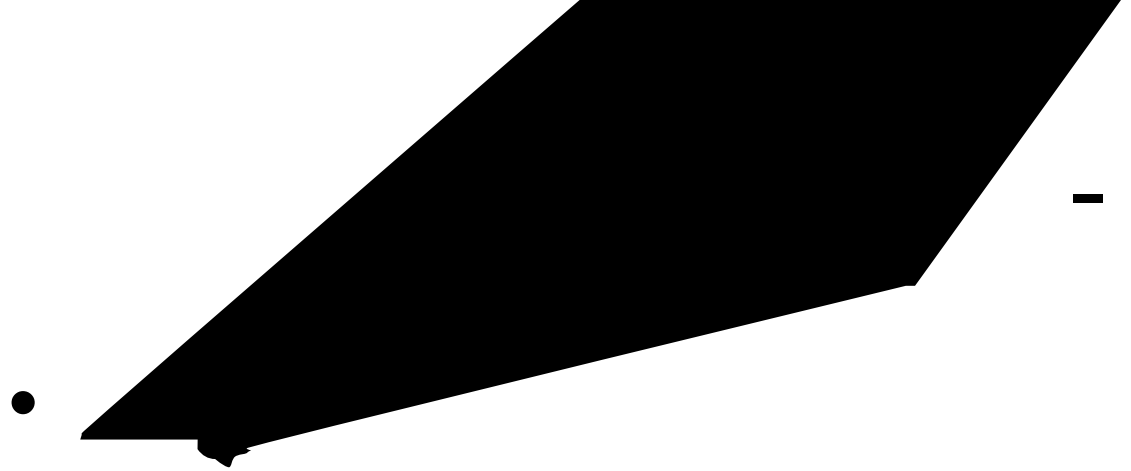
)

!

●

,

,

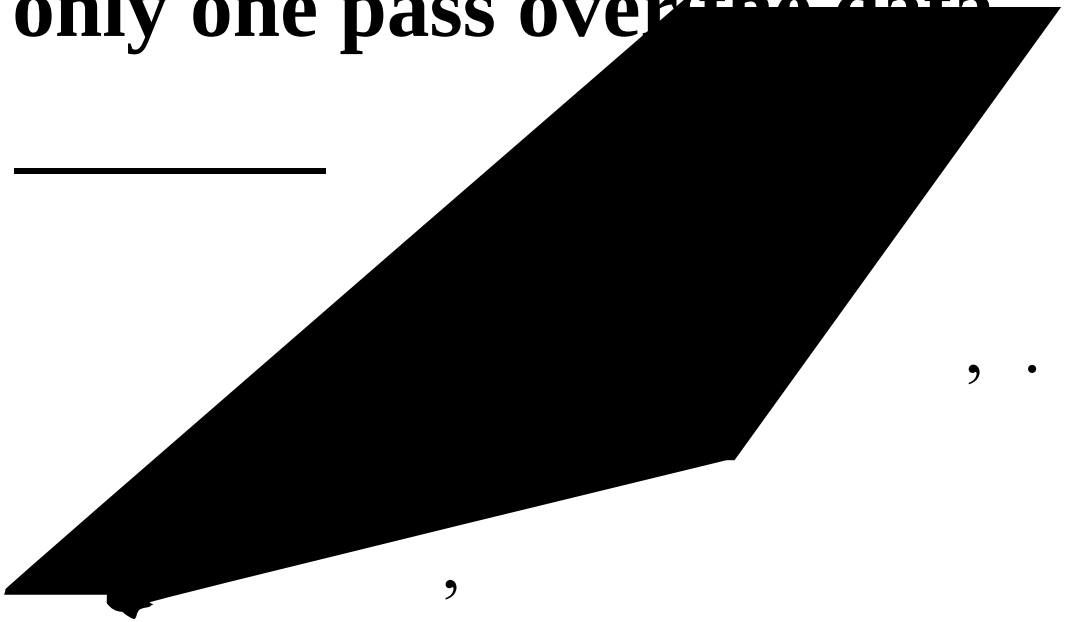


running estimate

– Effective exploitation of locality

& -

– **only one pass over the data**



, . .

.

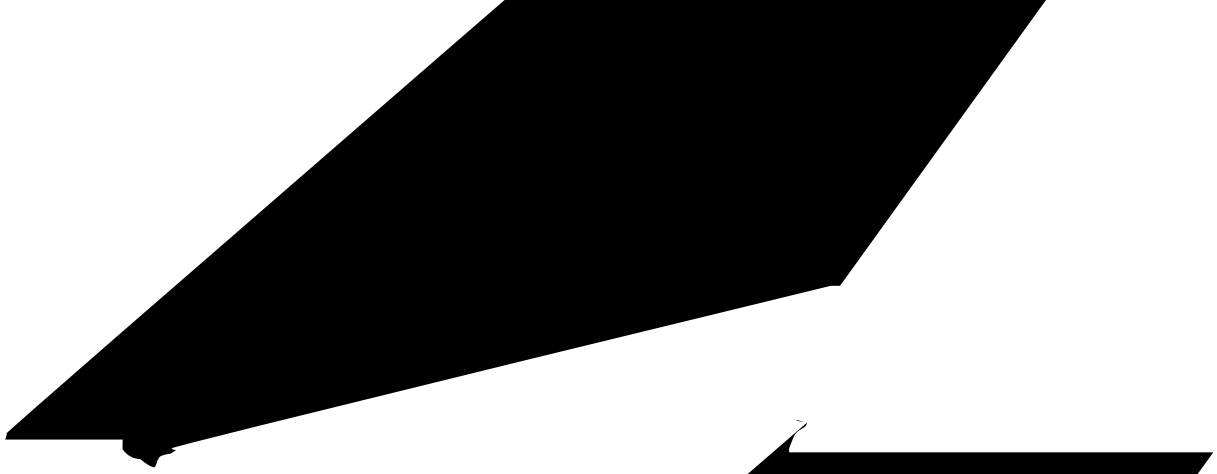
.

,

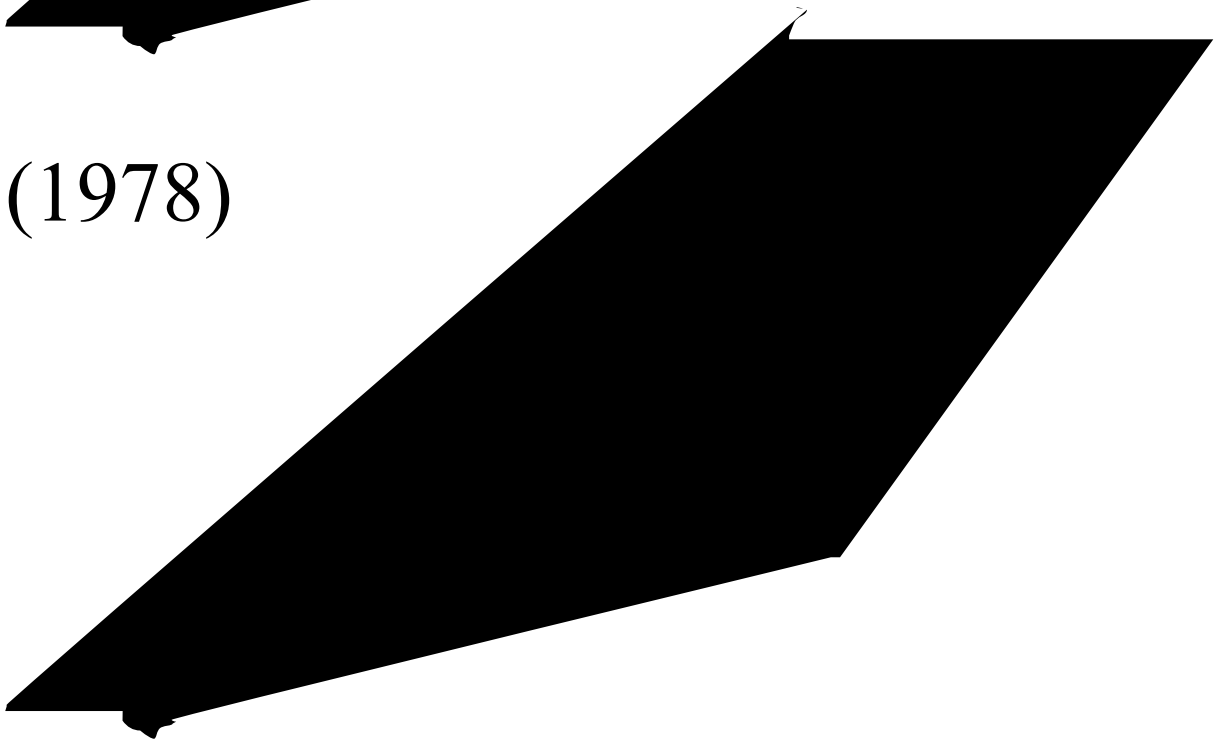
n *n*.

. . .

.

- 

(1978)

-
- 

(1987)

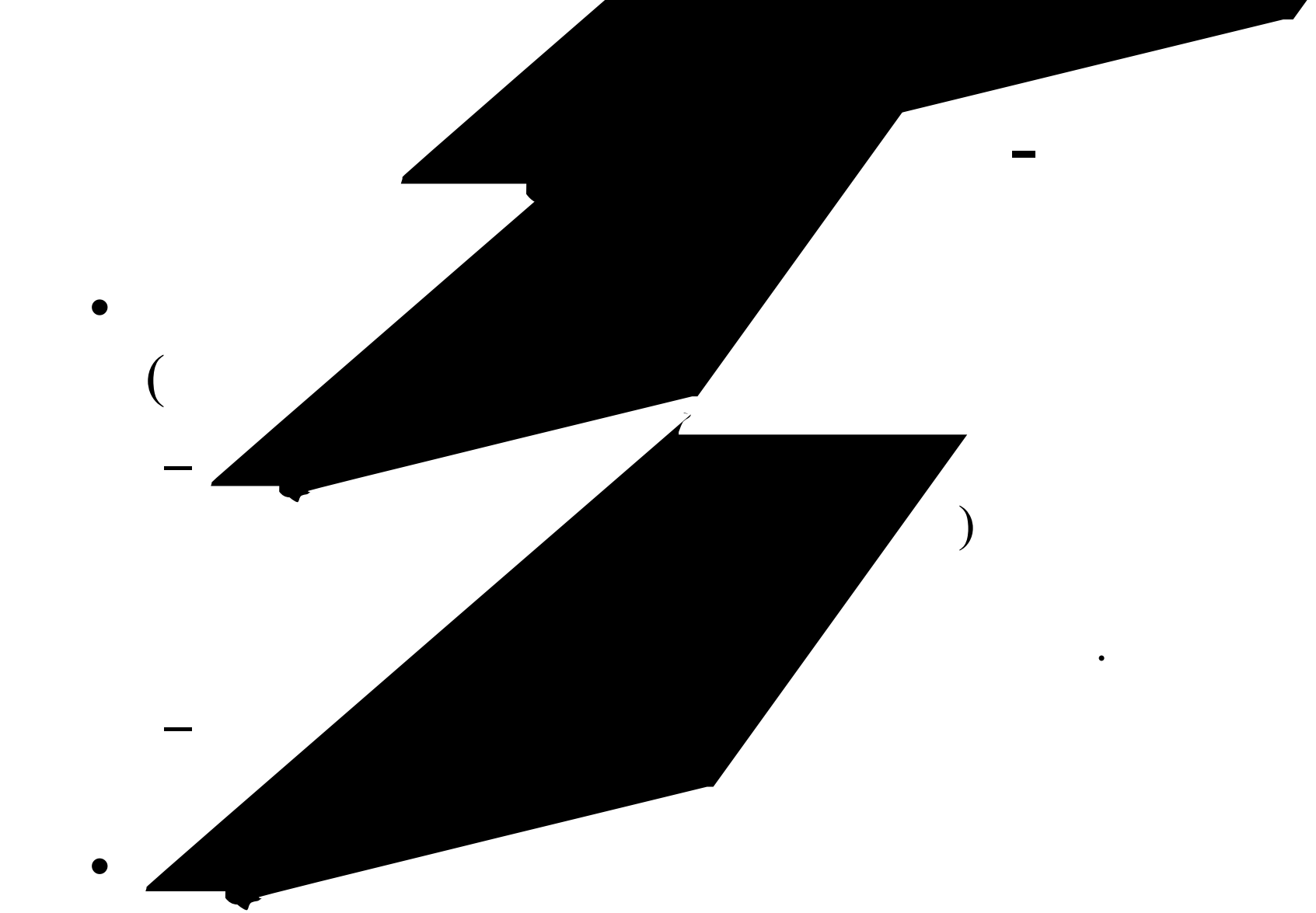
- ,

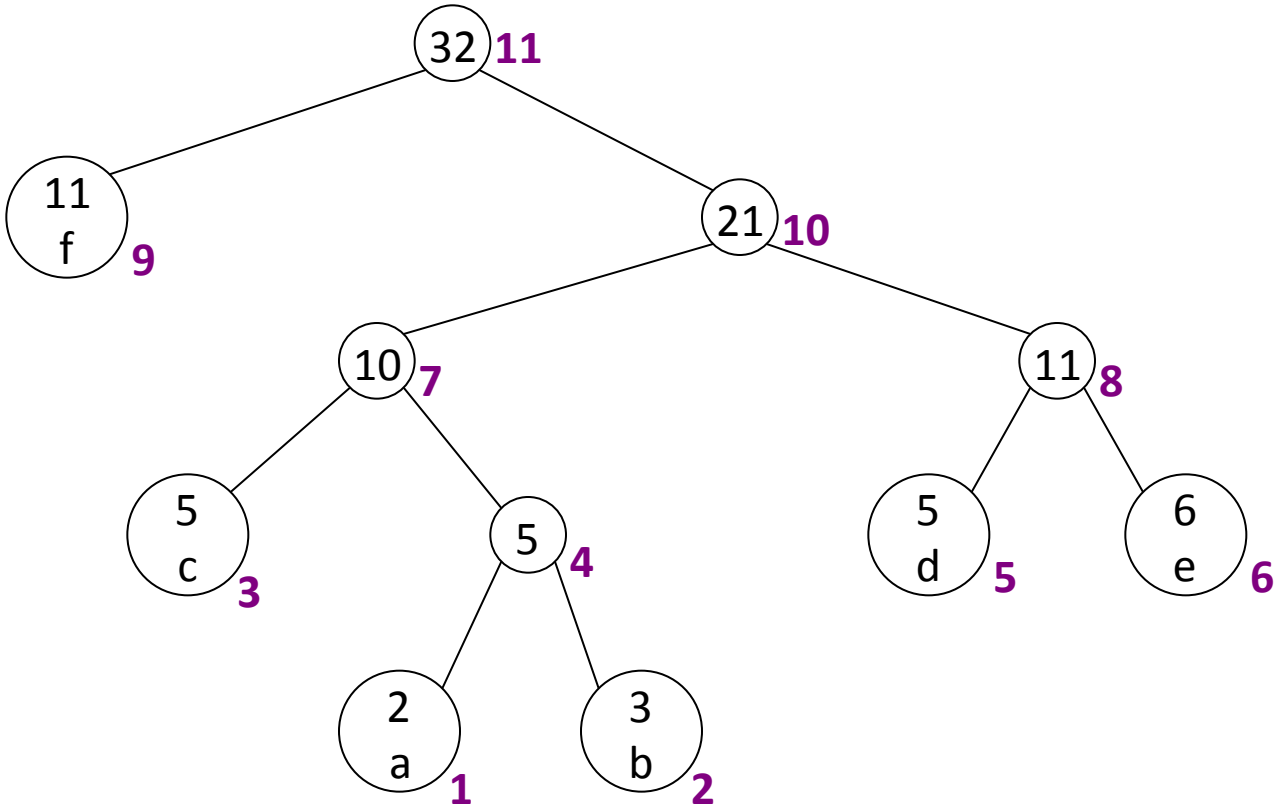
, . ., ,

- …

,

There is a contradiction?





- , 1 2, 3 4 ...

-

- ,

.

– *t*

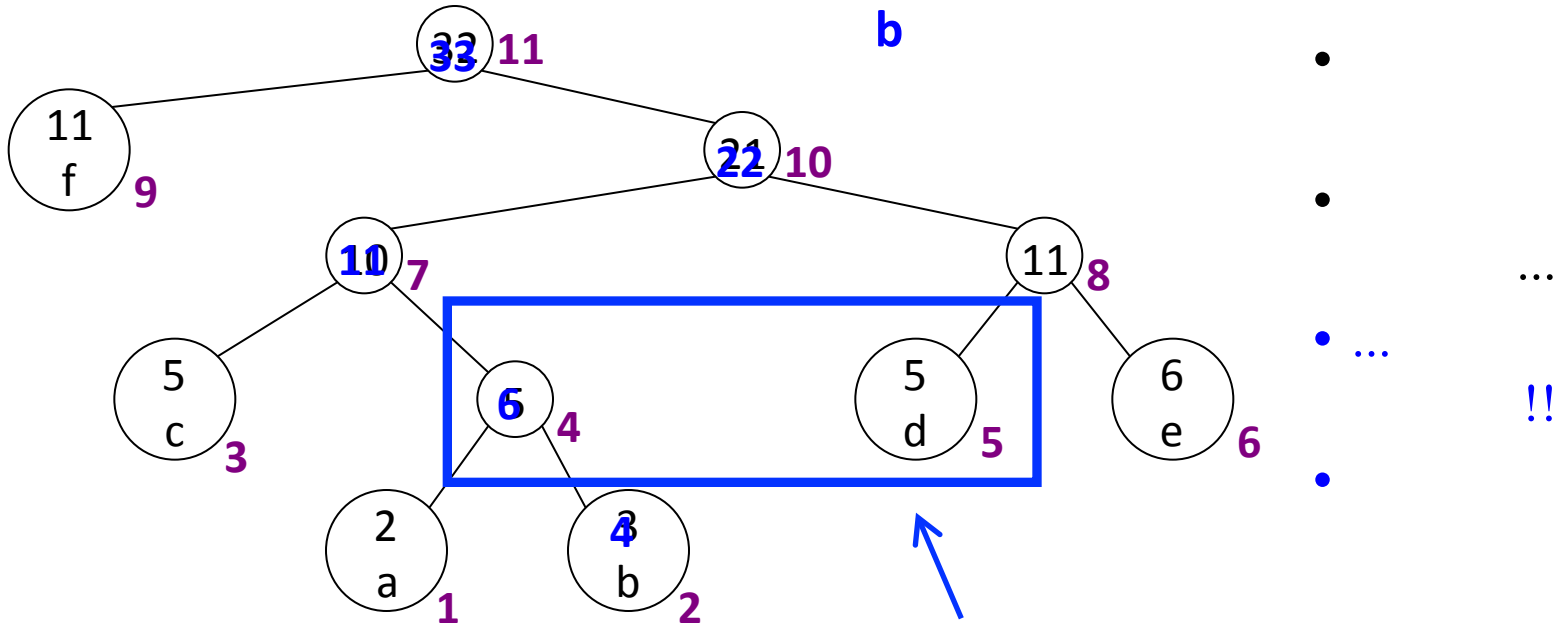
( )

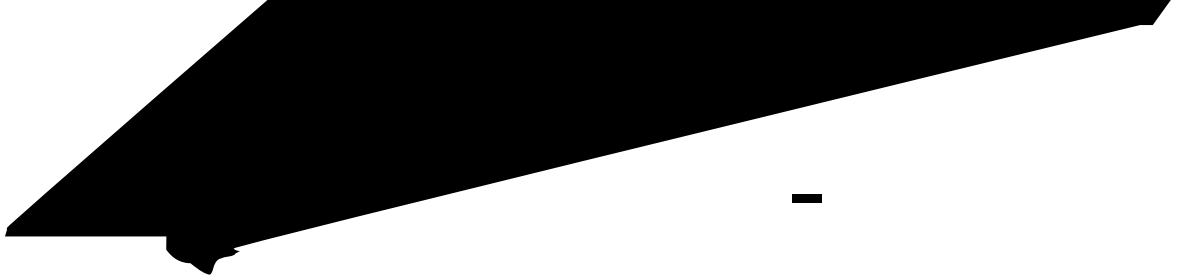
t+1

– *t*+1-

,

!


b
11
f
9
5
c
3
2
a
1
4
b
2
5
d
5
6
e
6
11
8
7
4
10
11
...
!!
no more ordered by
nondecreasing weight



-

- -

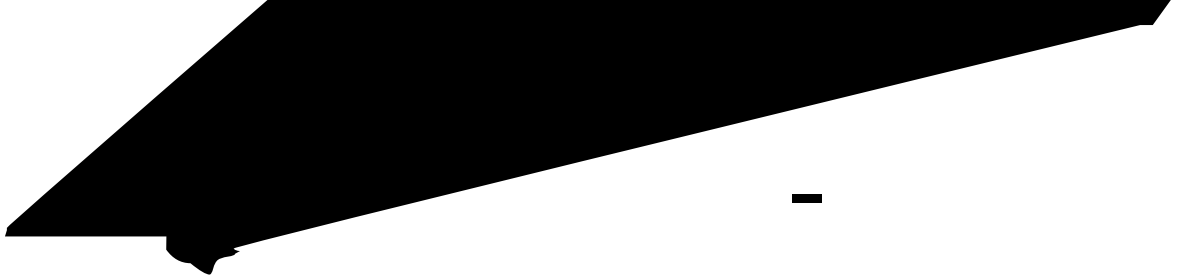
– :

t ,

simple increment

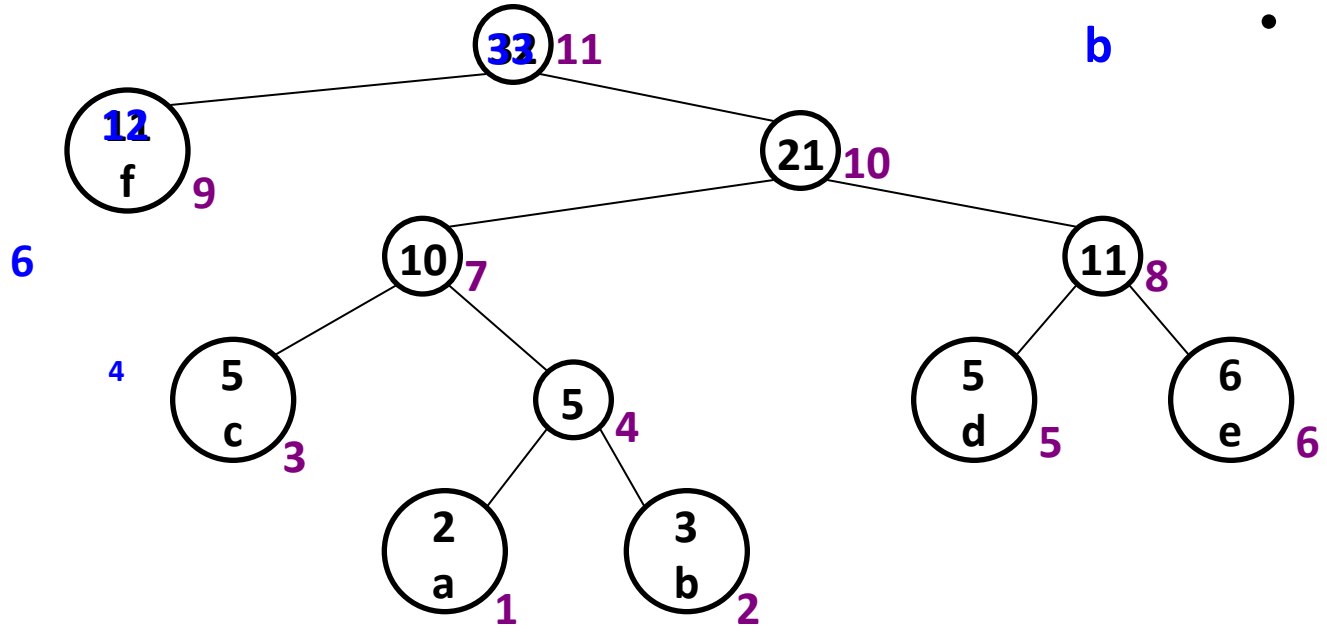
process can be applied succesfully (How?)

– : ,



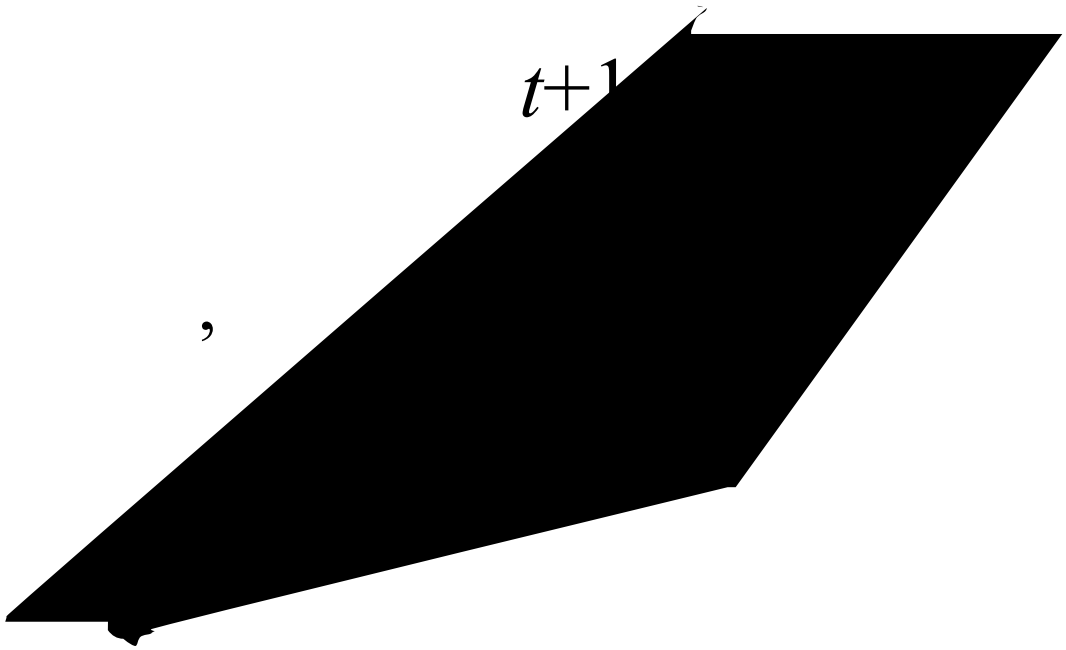
-

- t+1-
- ,
  ,
- —
-



- **First phase**
  - 2:
  - 4:
  - 5
  - 8:
  - 9
  - :
- **Second phase**

?

- 
t+1

-

•

.

?

-

- 0- 0)

.

de for the 0-node

and some h ich is the new symbol

– *n*

, *n* *n*

– ,

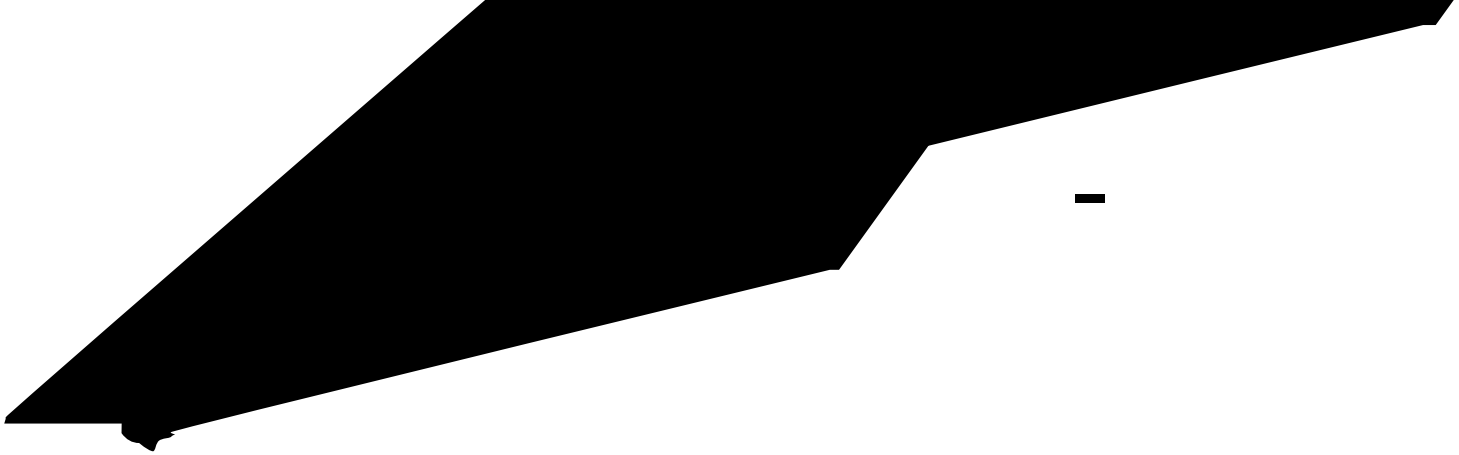
.

,

-

- 0- ,
  , ,
  1, 0-
-

- 0- . , ,
-
- ,
- 0-



,

(*compact*)

- **Exercise**
  -

e eae de eabe eae dcf

,

-

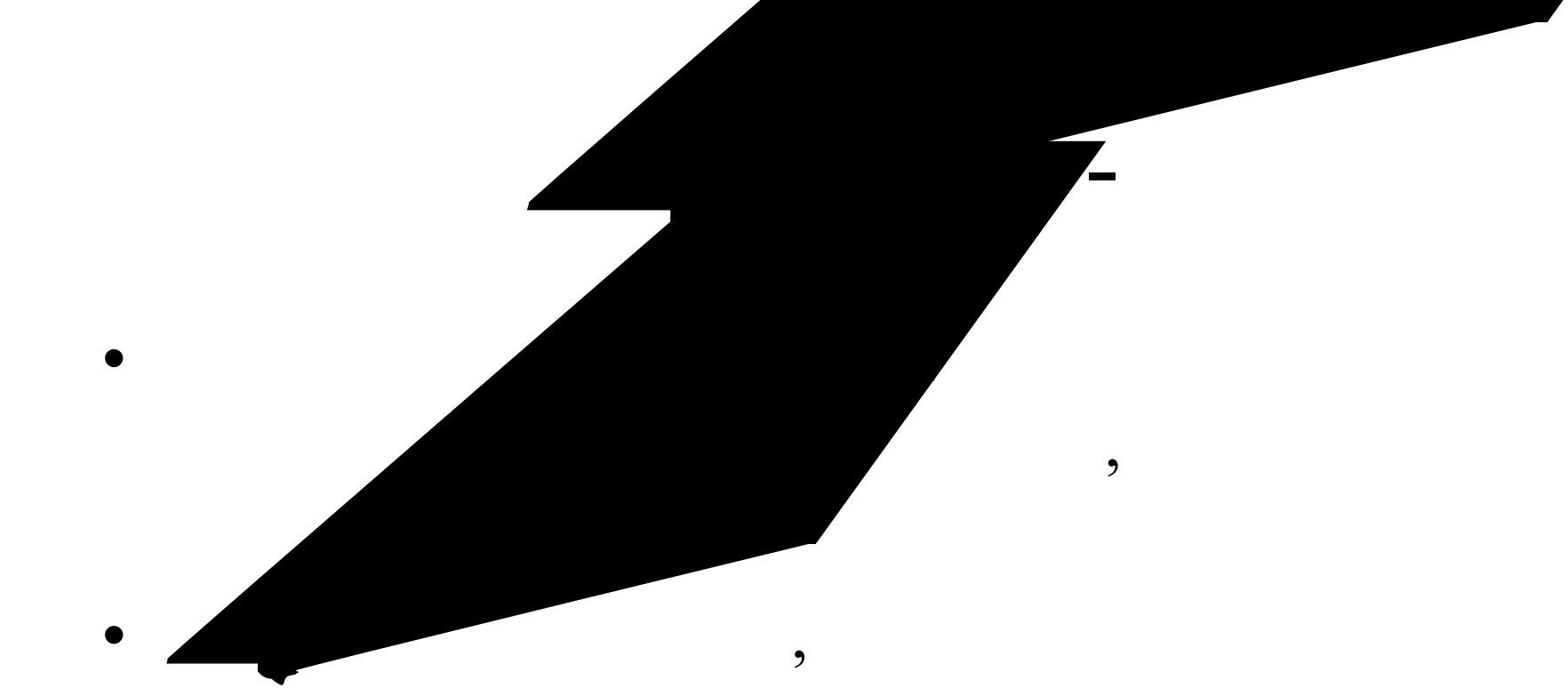
- *T*=

t *n* ,

$$- \quad +1 \leq \quad \leq 2 \quad + \quad -4 \quad +2$$

S (1987)

-



-

- –
(
) $\lceil d_t/2 \rceil$,
(
)
– , 1

-

- , $\sum_i w_i l_i$
, $\max_i l_i$. ,
, $\sum_i l_i$,

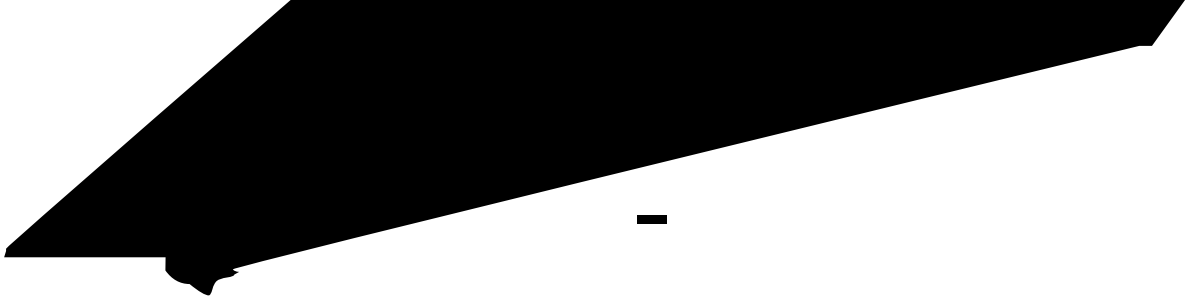
- , *implicit numbering*

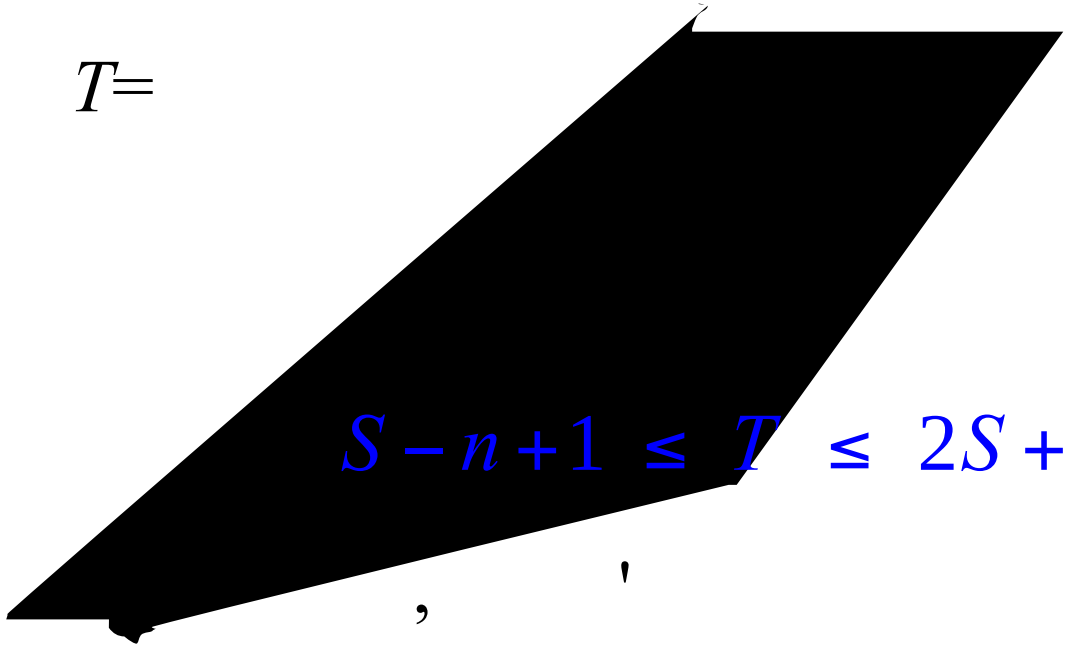
- ;
-
- (
),

- *invariant*

  – *for each weight w, all leaves of weight w precede (in the implicit numbering) all internal nodes of weight w*

- , , , ,



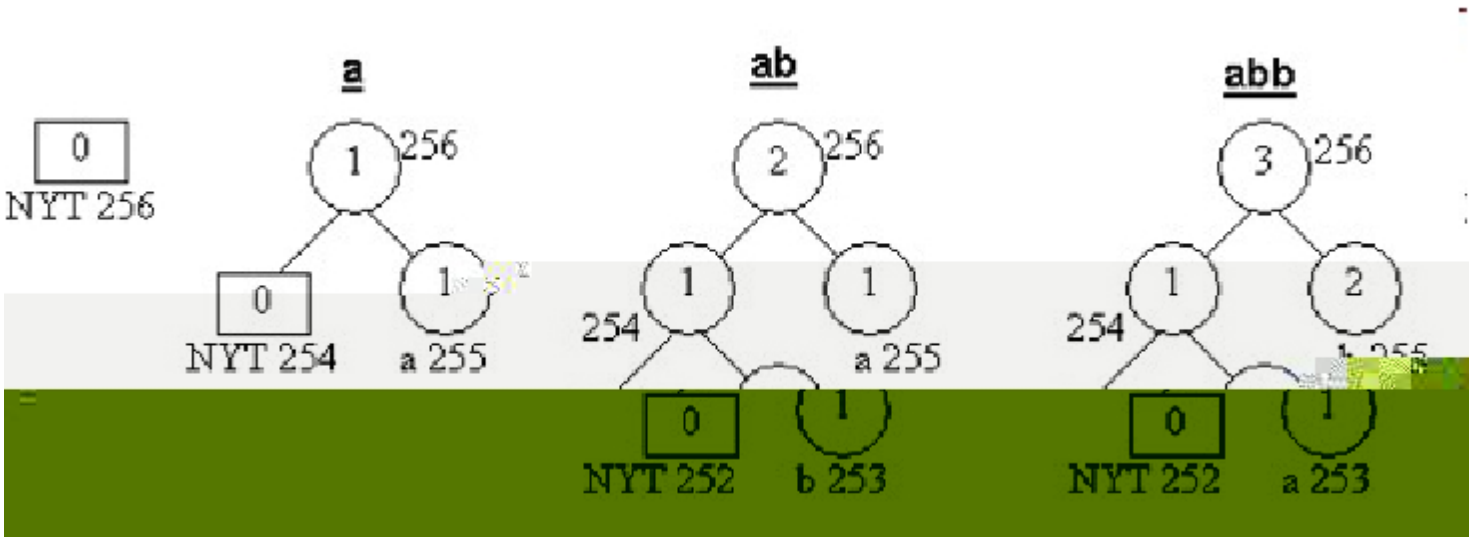
- $T=$  n ,

$$S - n + 1 \leq T \leq 2S + t - 2n + 1$$

- , '
- ,

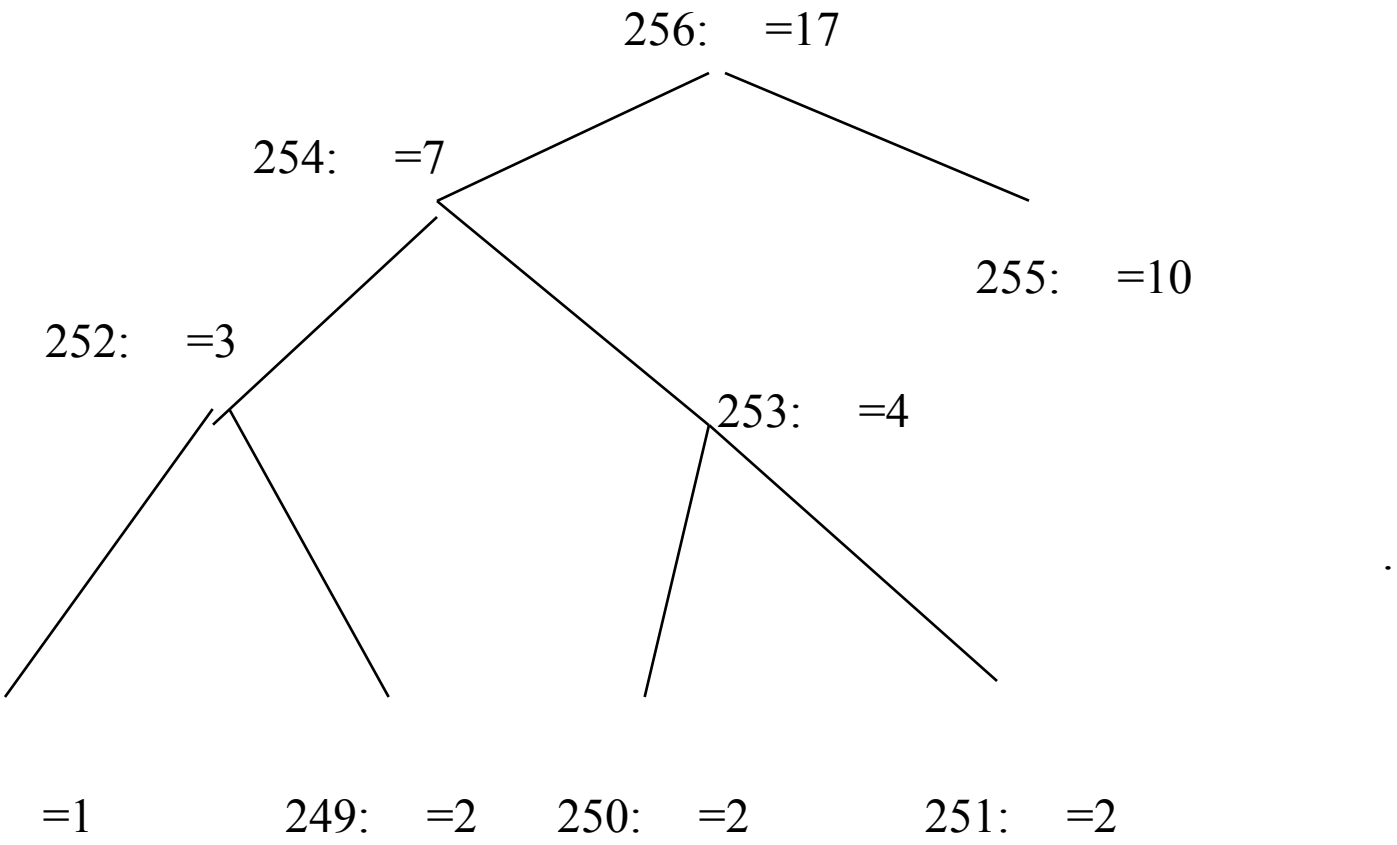
: 01100001011000100110001001100010011000100110001001100001

: 01100001**0**01100010**0111101**

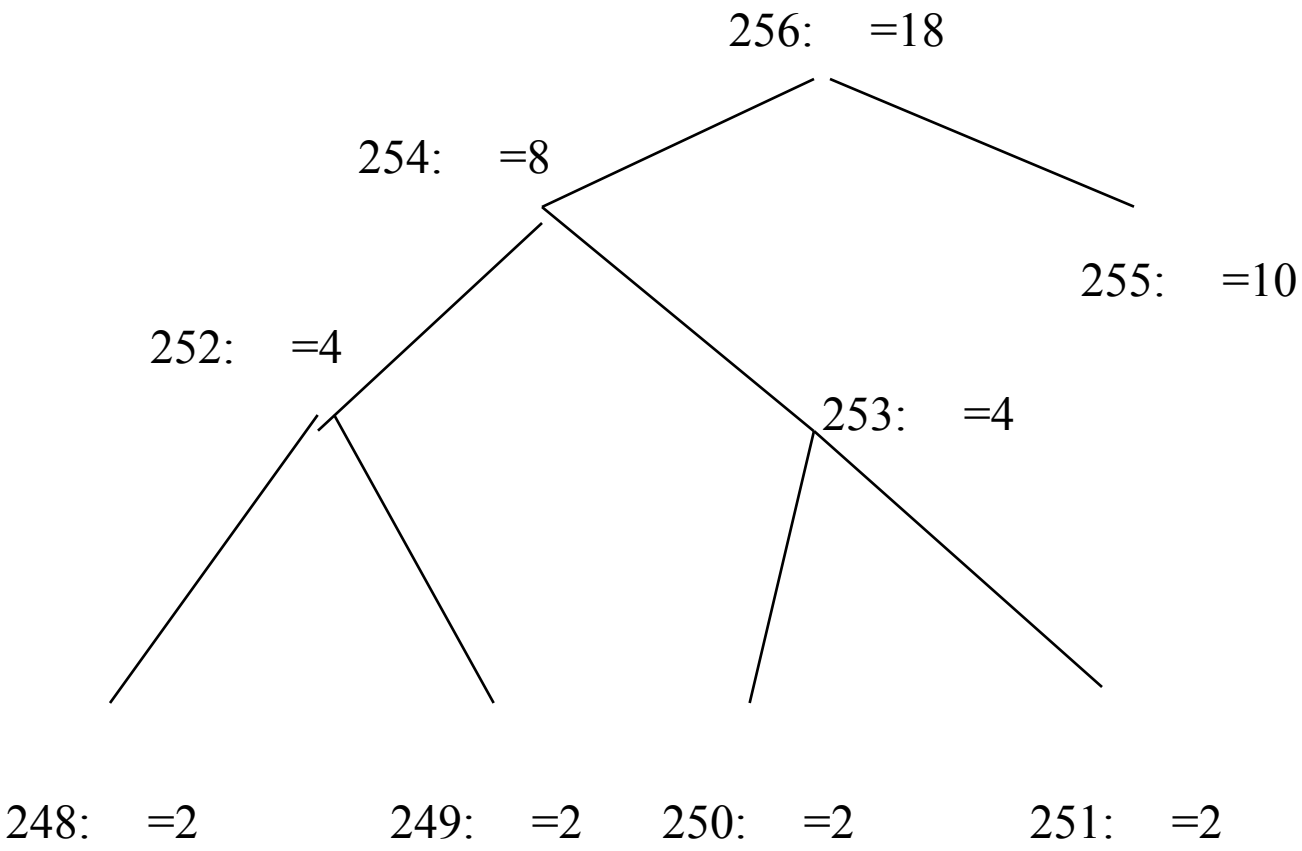


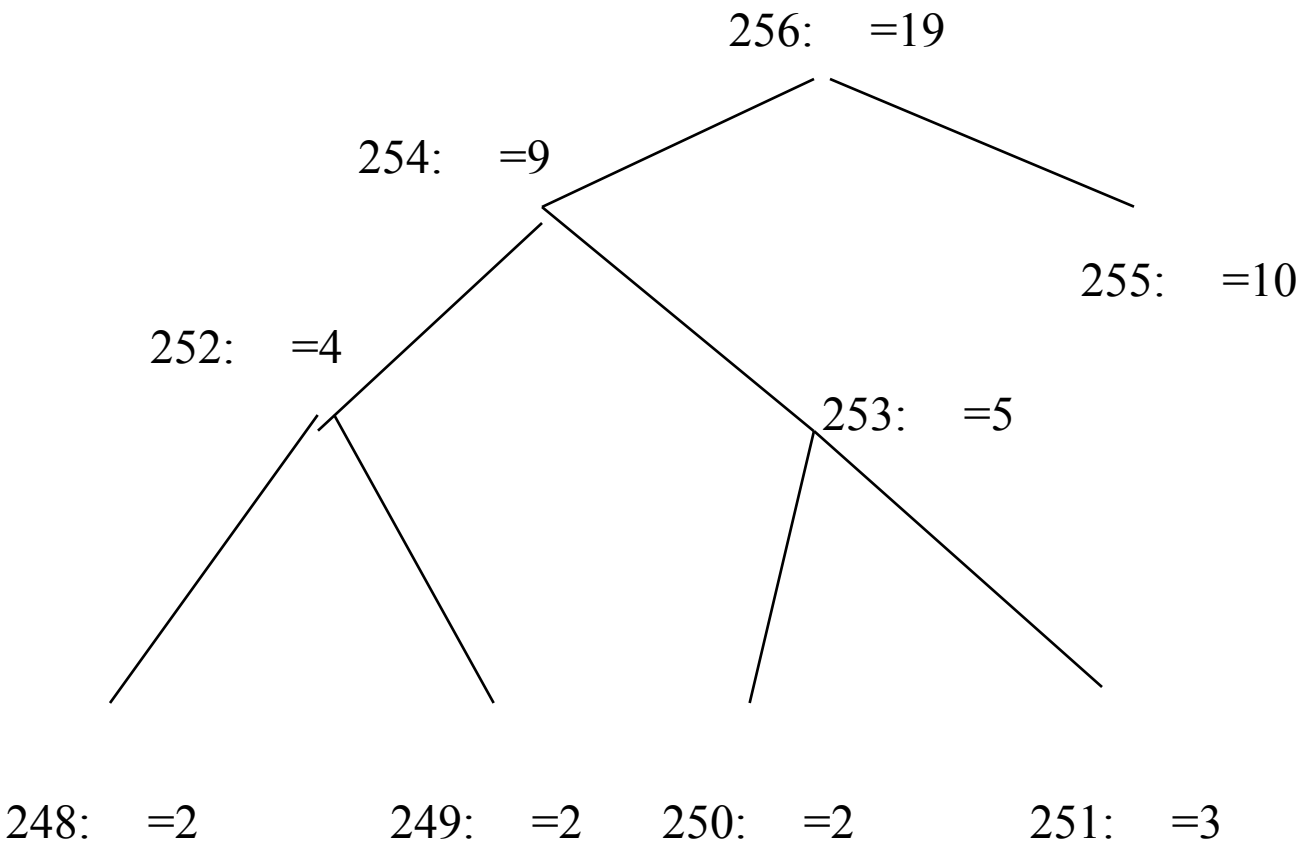
: 01100001

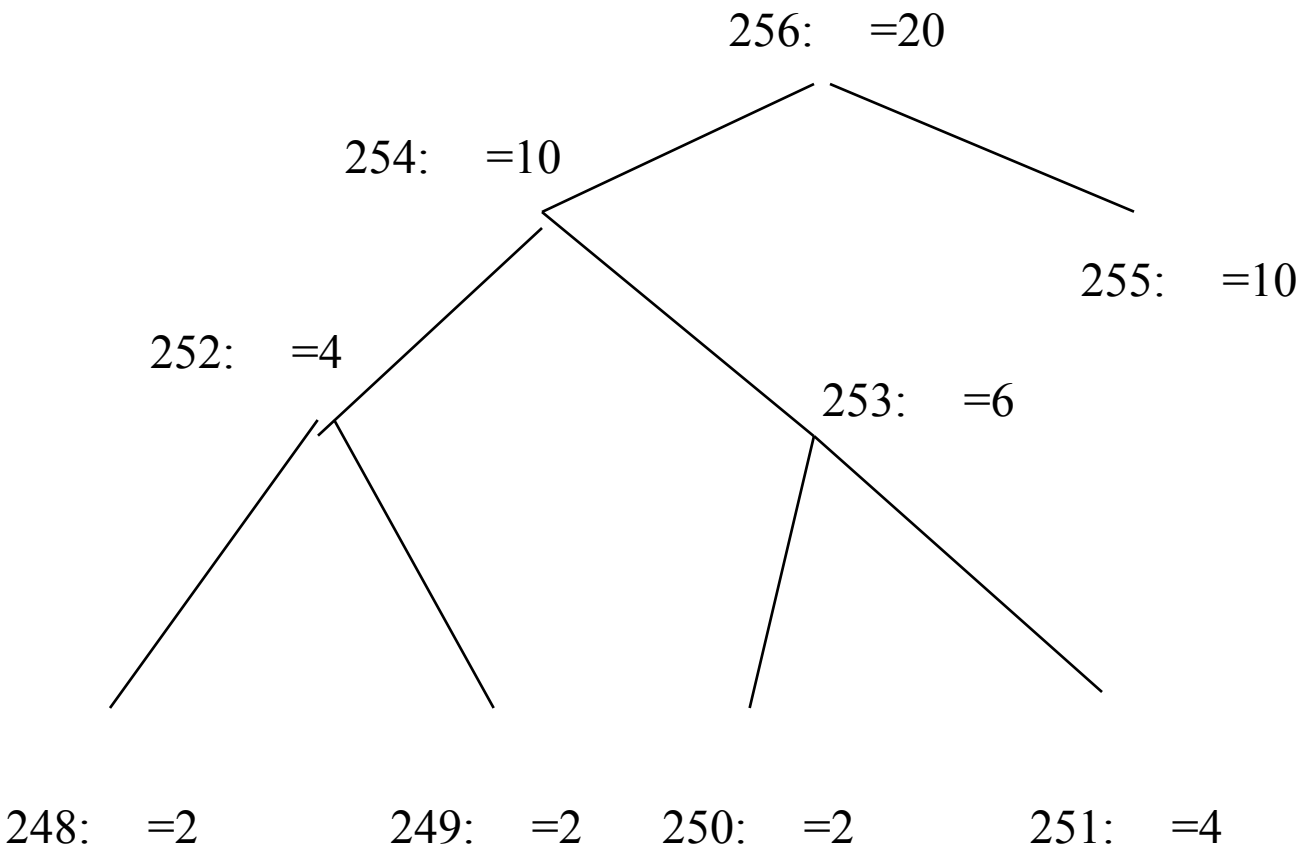
: 01100010

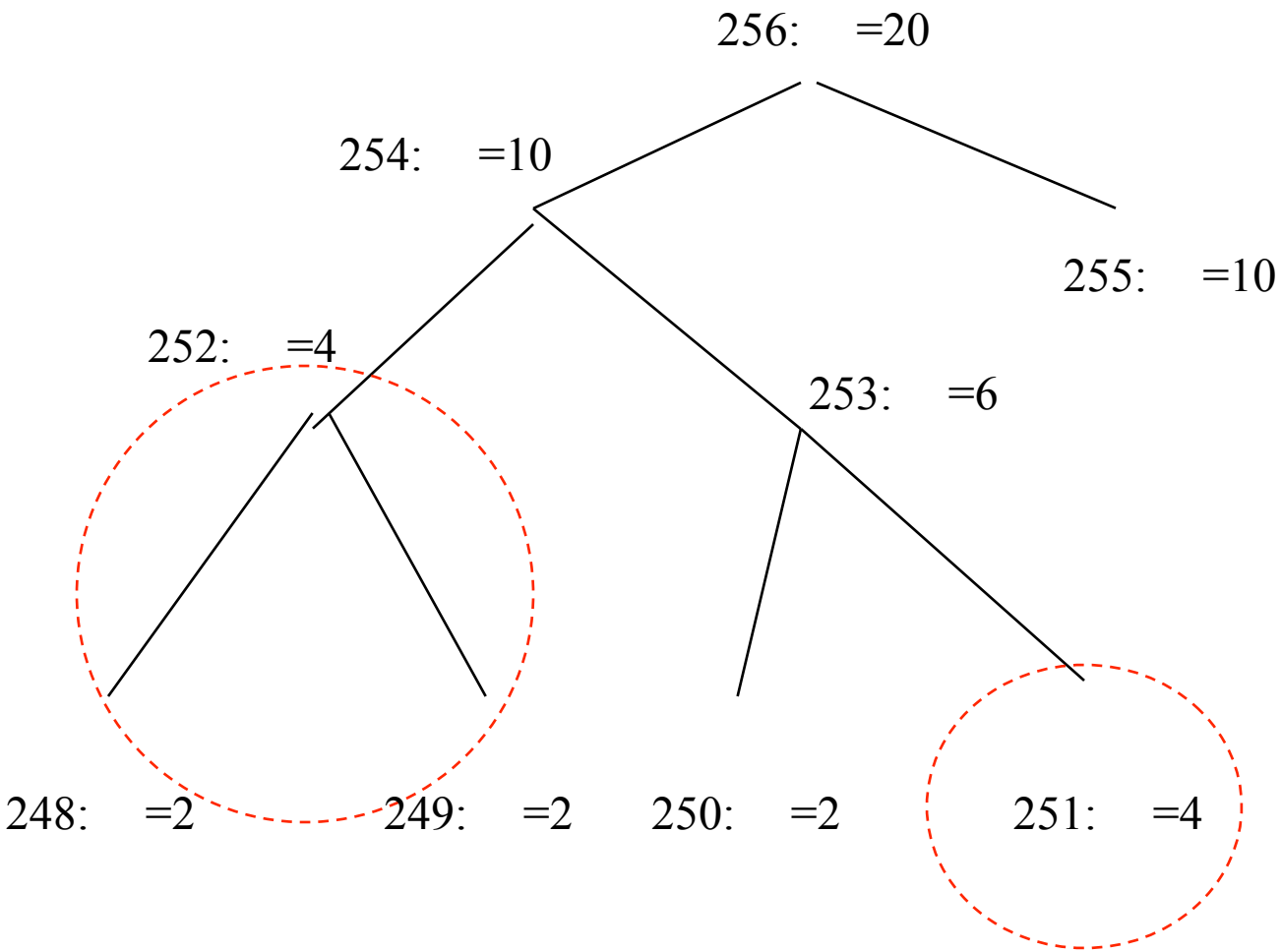

256: =17
254: =7
255: =10
252: =3
253: =4
248: =1
249: =2
250: =2
251: =2

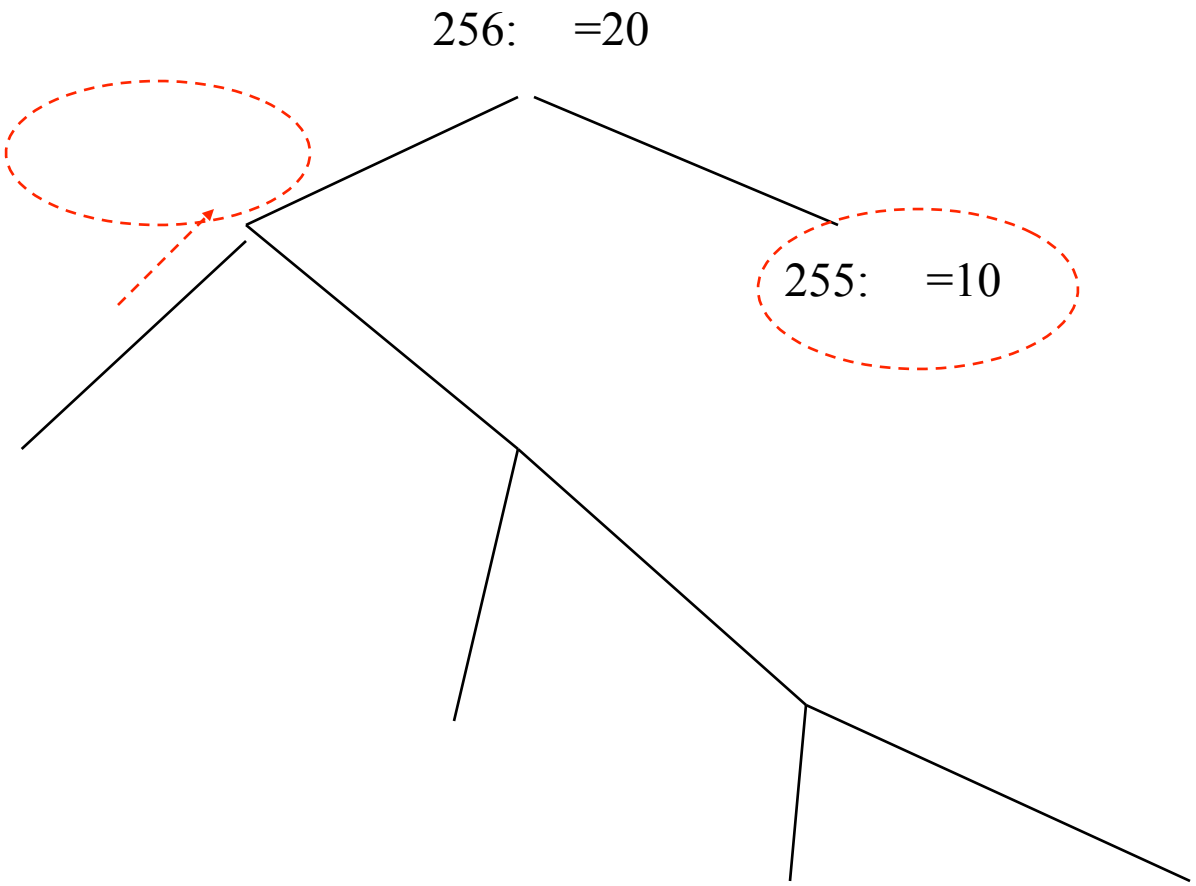
.

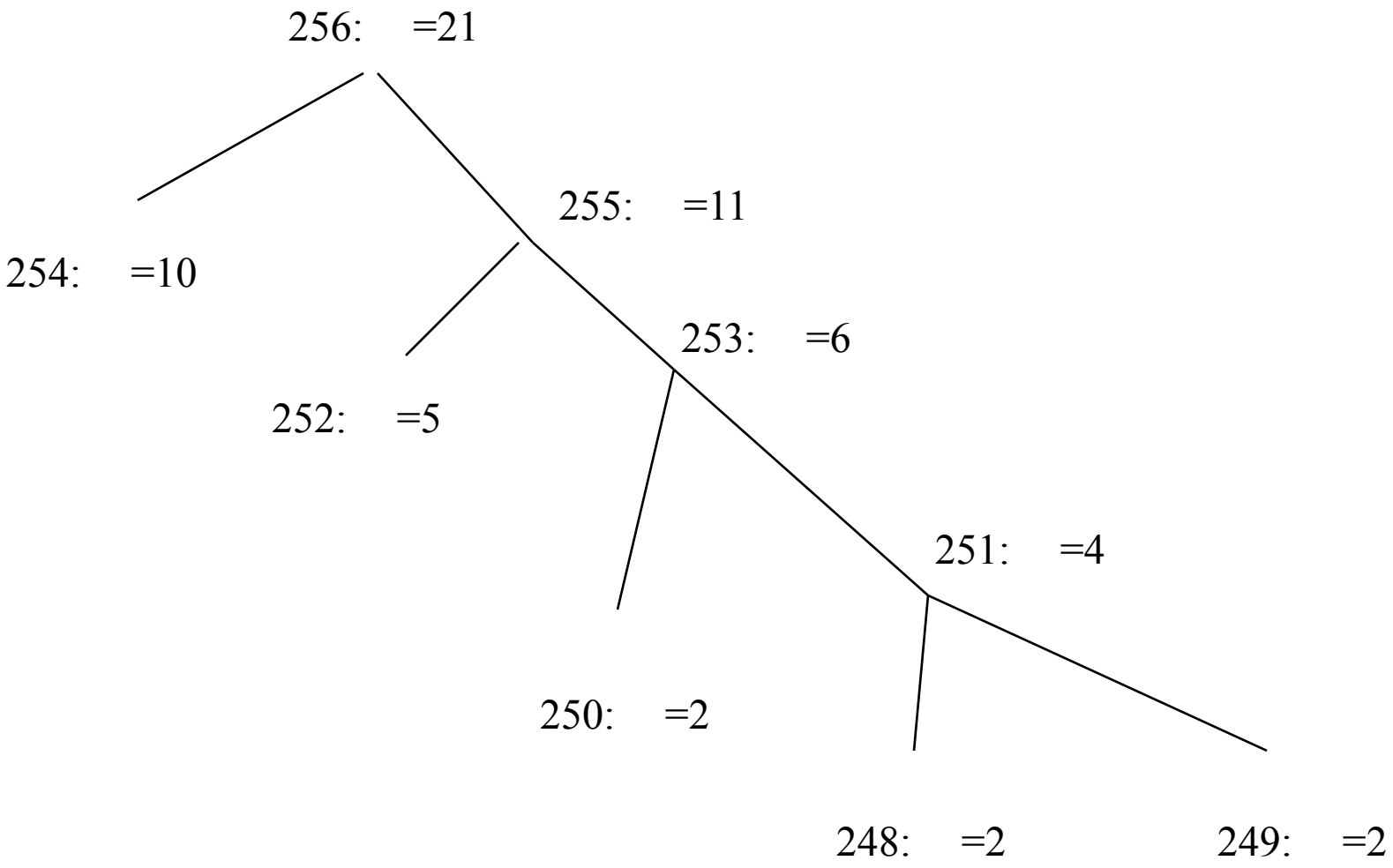

256: =18
254: =8
255: =10
252: =4
253: =4
248: =2
249: =2
250: =2
251: =2


256: =19
254: =9
255: =10
252: =4
253: =5
248: =2
249: =2
250: =2
251: =3


256: =20
254: =10
255: =10
252: =4
253: =6
248: =2
249: =2
250: =2
251: =4


256: =20
254: =10
255: =10
252: =4
253: =6
248: =2
249: =2
250: =2
251: =4


256: =20
255: =10


256: =21
255: =11
254: =10
253: =6
252: =5
251: =4
250: =2
248: =2
249: =2

(.)

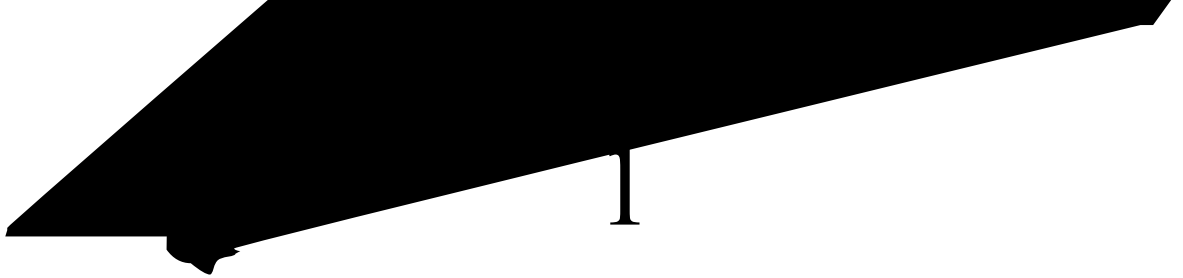
. ., (

)

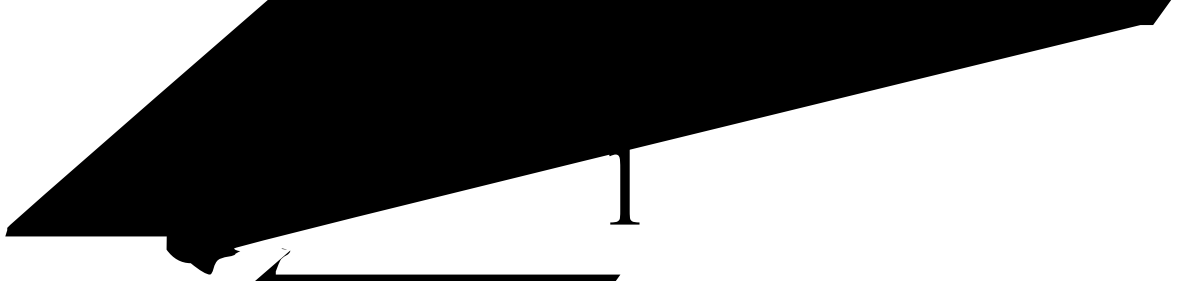
.

(.

)



- **Size Determination of Huffman & LZW Encoded File**
  - (&)
    -8
    .
  -
    - . . +



–

–

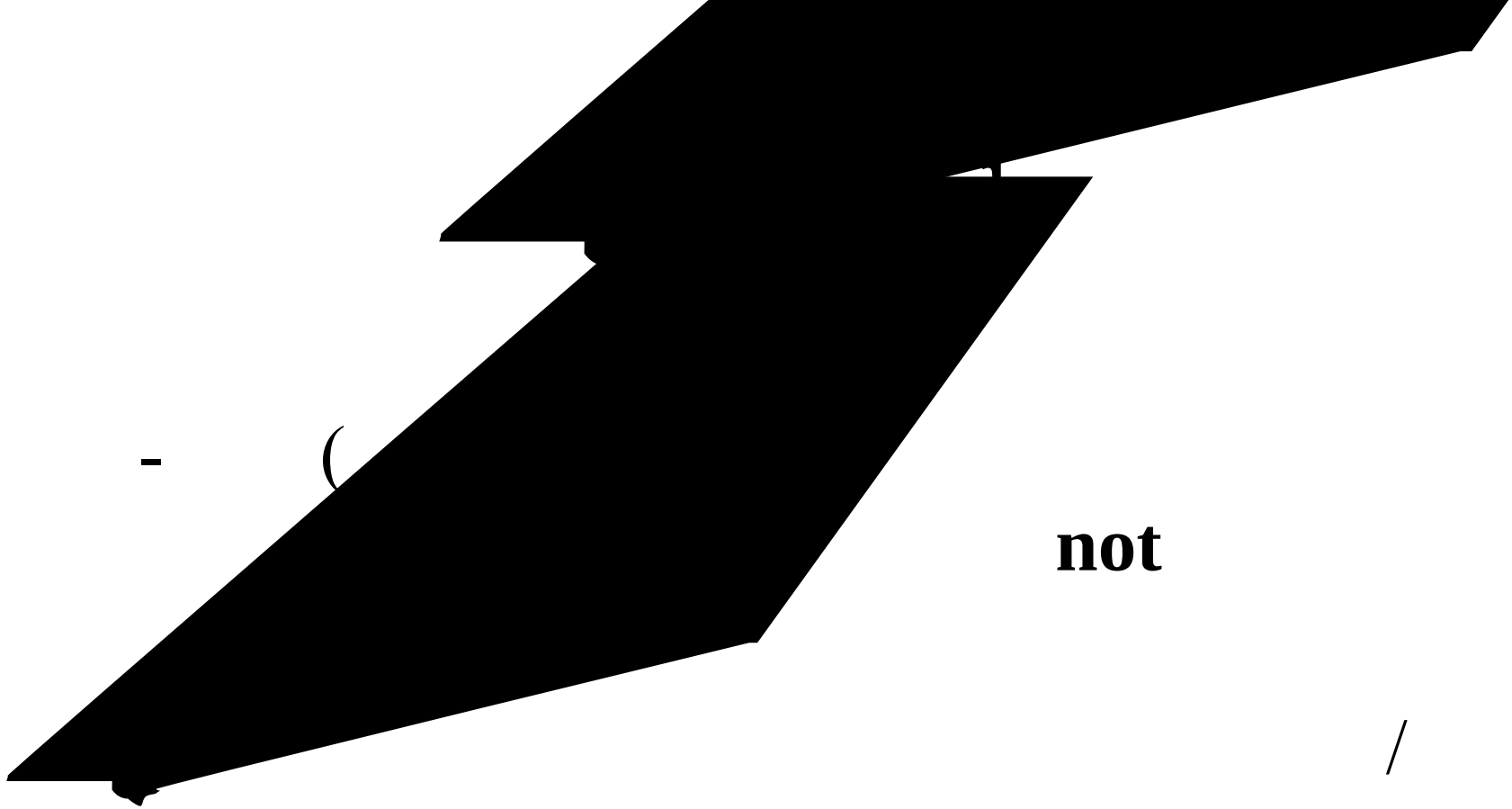
–

–

9 20

( )

( )



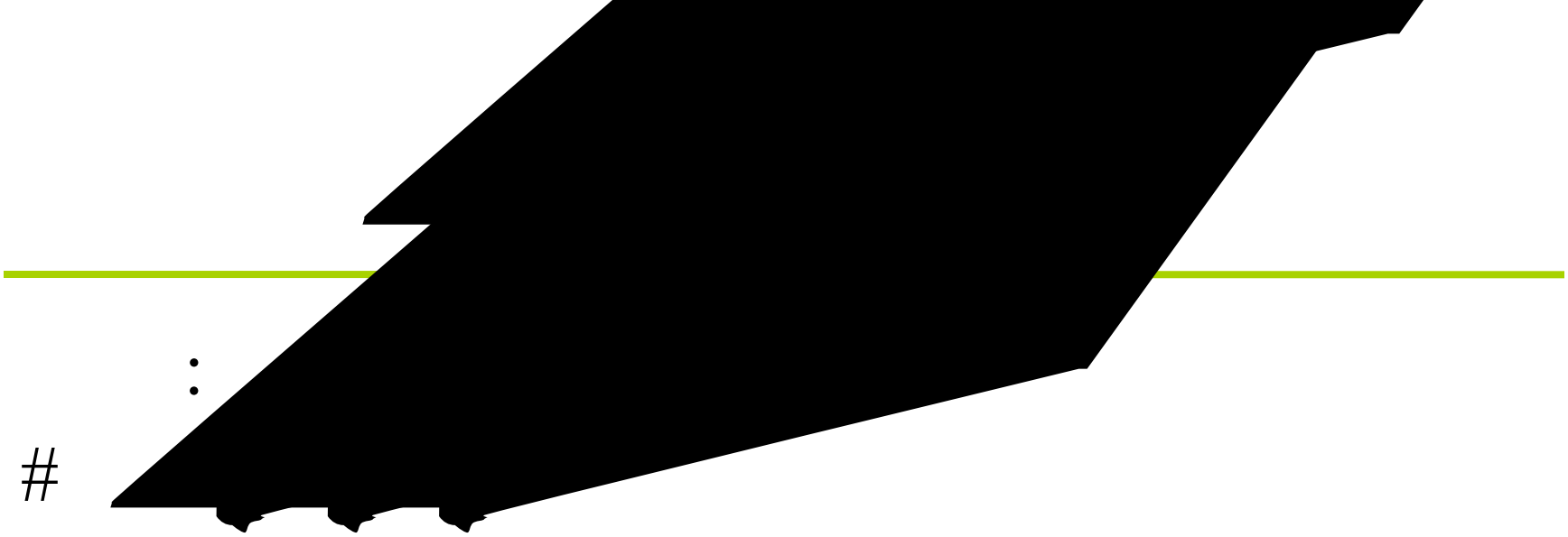
- -
- /
++ , - ,
, **zero**
.

1

$

$

4 3 10 5 $



#BANANAS
S#BANANA
AS#BANAN
NAS#BANA
ANAS#BAN
NANAS#BA
ANANAS#B
BANANAS#

```
#BANANAS
ANANAS#B
ANAS#BAN
AS#BANAN
BANANAS#
NANAS#BA
NAS#BANA
S#BANANA
```

#BANANAS
ANANAS#B
ANAS#BAN
AS#BANAN
BANANAS#
NANAS#BA
NAS#BANA
S#BANANA

:

$

$

:

S

B

N

N

#

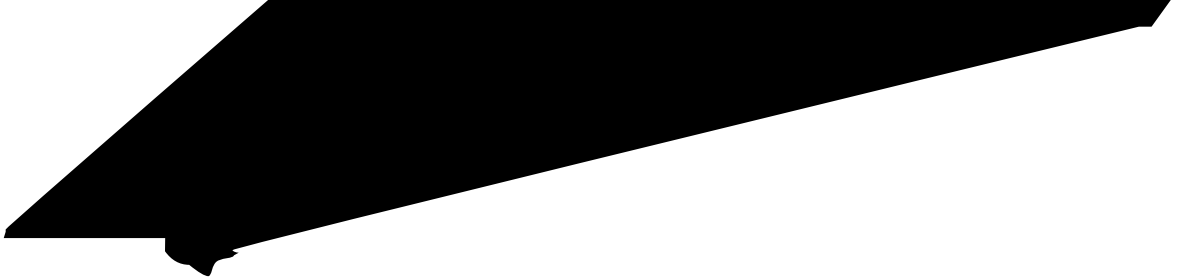
A

A

A

S
B
N
N
#
A
A
A

#AAABNNS



S#
BA
NA
NA
#B
AN
AN
AS

#B
AN
AN
AS
BA
NA
NA
S#

S#B
BAN
NAN
NAS
#BA
ANA
ANA
AS#

```
#BA
ANA
ANA
AS#
BAN
NAN
NAS
S#B
```

S#BA
BANA
NANA
NAS#
#BAN
ANAN
ANAS
AS#B

```
#BAN
ANAN
ANAS
AS#B
BANA
NANA
NAS#
S#BA
```

S#BAN
BANAN
NANAS
NAS#B
#BANA
ANANA
ANAS#
AS#BA

```
#BANA
ANANA
ANAS#
AS#BA
BANAN
NANAS
NAS#B
S#BAN
```

S#BANA
BANANA
NANAS#
NAS#BA
#BANAN
ANANAS
ANAS#B
AS#BAN

```
#BANAN
ANANAS
ANAS#B
AS#BAN
BANANA
NANAS#
NAS#BA
S#BANA
```

S#BANAN
BANANAS
NANAS#B
NAS#BAN
#BANANA
ANANAS#
ANAS#BA
AS#BANA

```
#BANANA
ANANAS#
ANAS#BA
AS#BANA
BANANAS
NANAS#B
NAS#BAN
S#BANAN
```

S#BANANA
BANANAS#
NANAS#BA
NAS#BANA
#BANANAS
ANANAS#B
ANAS#BAN
AS#BANAN

(?)

#BANANAS
ANANAS#B
ANAS#BAN
AS#BANAN
BANANAS#
NANAS#BA
NAS#BANA
S#BANANA

( )

function ( )

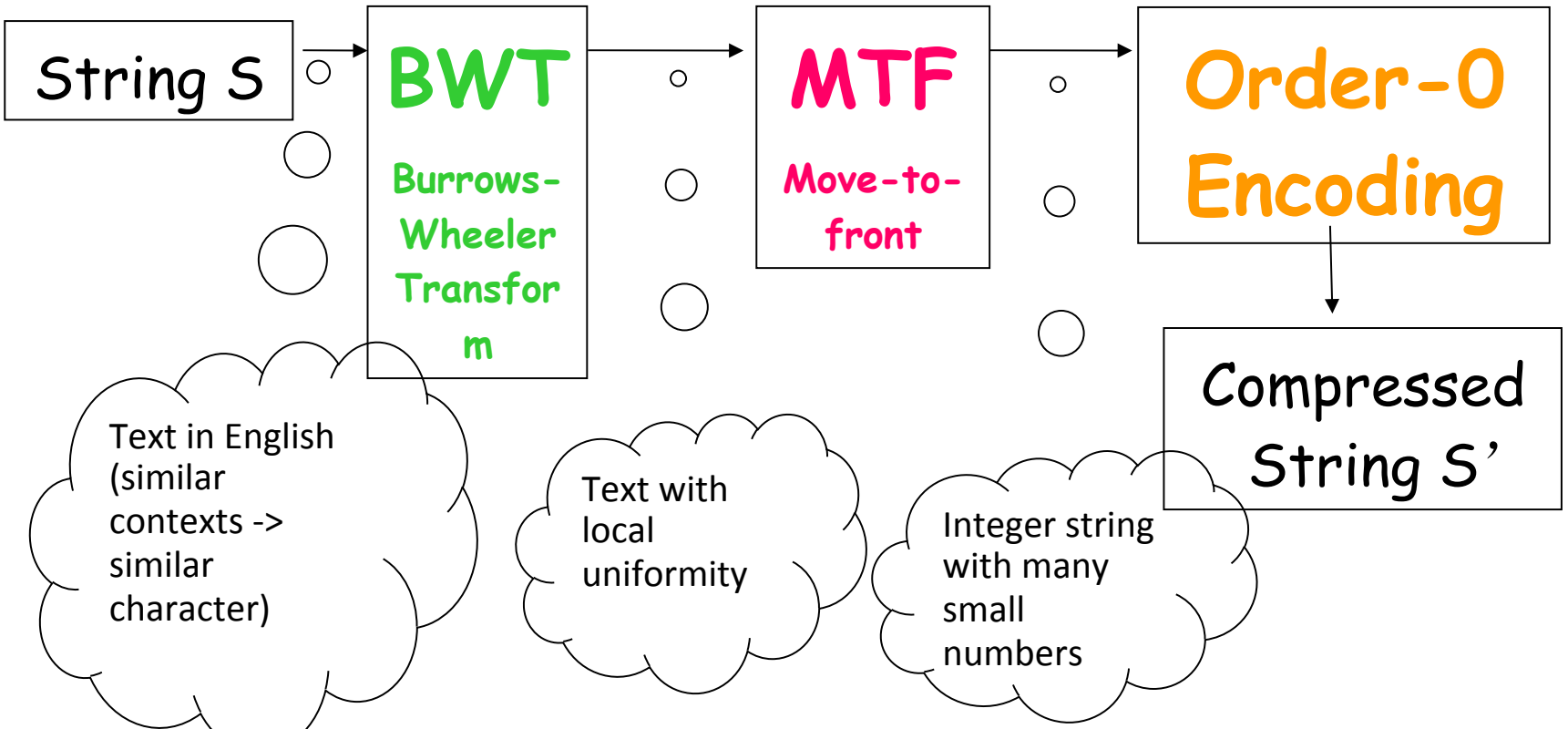
repeat ( ) **times**

//

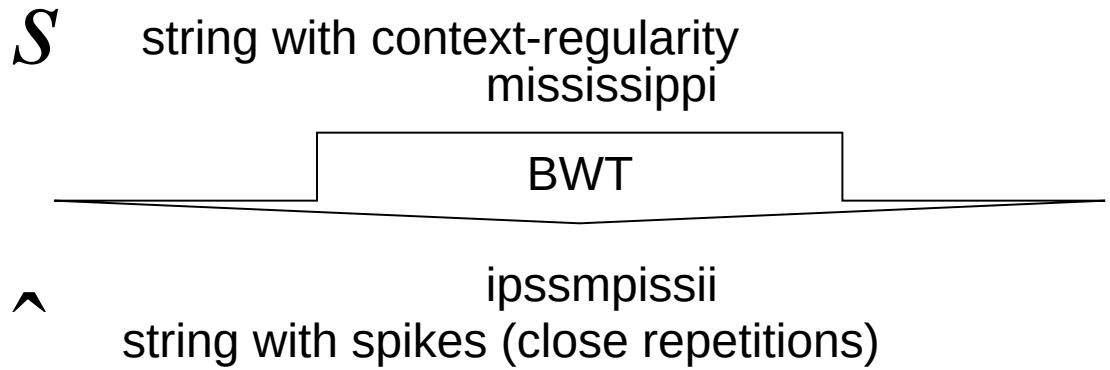
return (' ')

BW0

The Main Burrows-Wheeler Compression Algorithm:

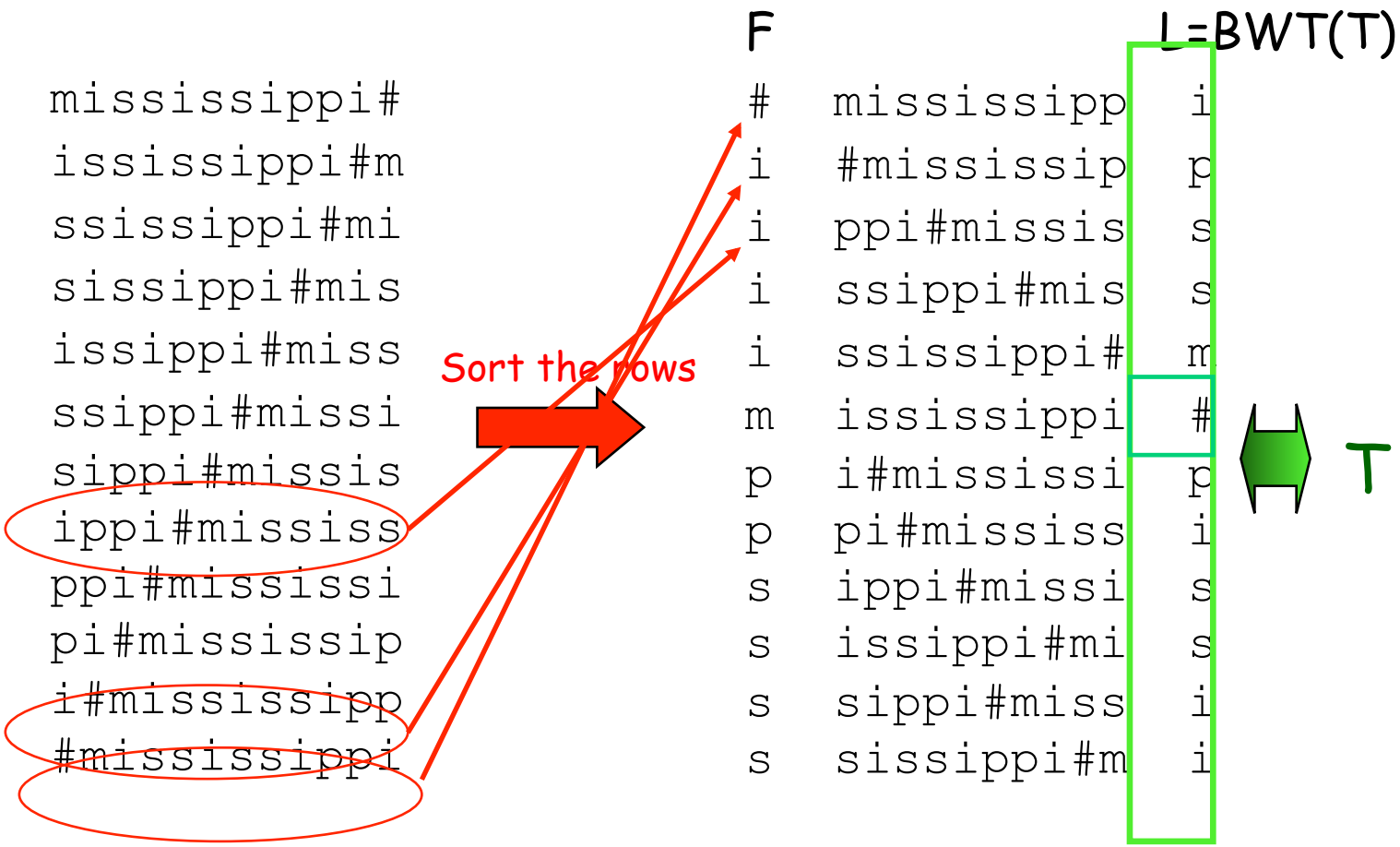


The BWT

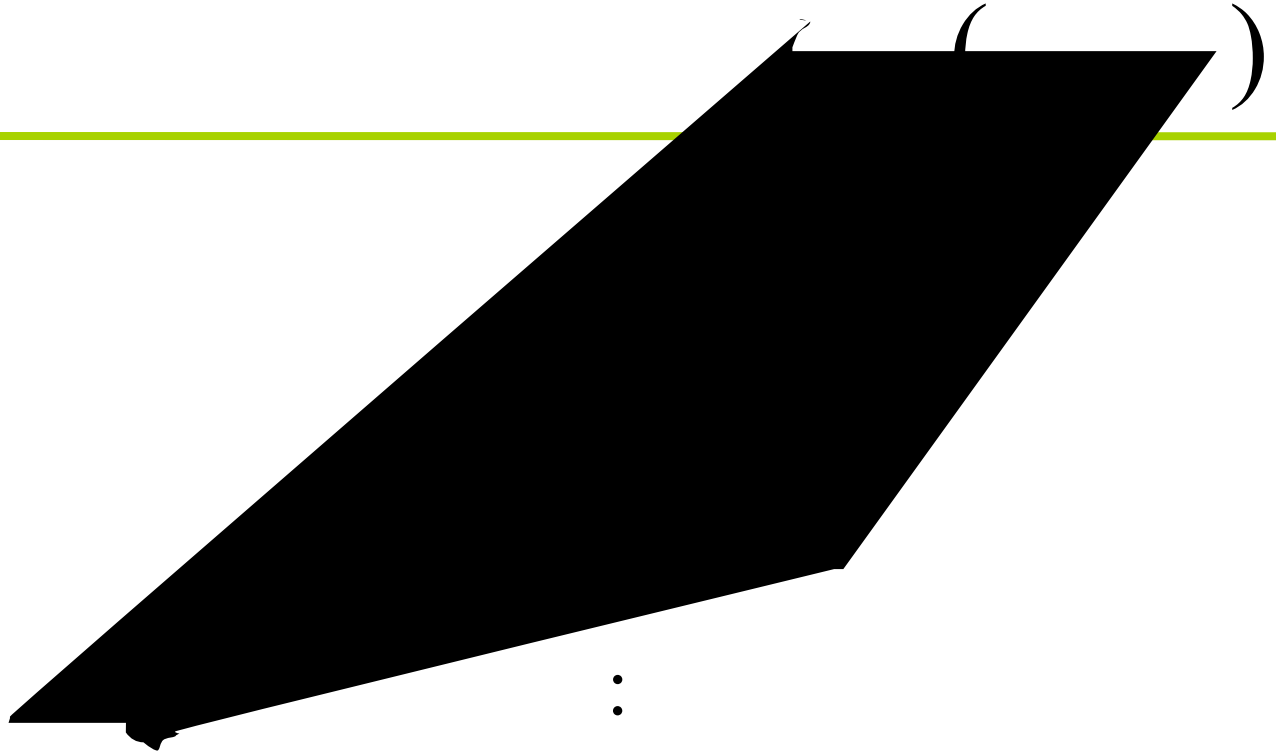

S
string with context-regularity
mississippi
BWT
ipssmpissii
$\hat{}$
string with spikes (close repetitions)

The BWT

T = mississippi#



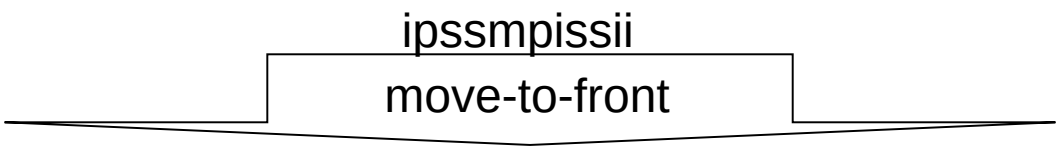
BWT sorts the characters by their post-context



Move To Front

- By Bentley, Sleator, Tarjan and Wei（’86)

$\hat{}$ string with spikes (close repetitions)



0,0,0,0,0,2,4,3,0,1,0

S' integer string with small numbers

Move to Front

abracadabra		**a**,b,r,c,d

Move to Front

abracadabra		**a**,b,r,c,d
a**b**racadabra	0	a,**b**,r,c,d

Move to Front

abracadabra		**a**,b,r,c,d
a**b**racadabra	0	a,**b**,r,c,d
ab**r**acadabra	0,1	b,a,**r**,c,d

Move to Front

abracadabra		**a**,b,r,c,d
a**b**racadabra	0	a,**b**,r,c,d
ab**r**acadabra	0,1	b,a,**r**,c,d
abr**a**cadabra	0,1,2	r,b,**a**,c,d

Move to Front

abr**a**cadabra	0,1,2	r,b,**a**,c,d
abra**c**adabra	0,1,2,2	a,r,b,**c**,d

Move to Front

abracadabra		**a**,b,r,c,d
a**b**racadabra	0	a,**b**,r,c,d
ab**r**acadabra	0,1	b,a,**r**,c,d
abr**a**cadabra	0,1,2	r,b,**a**,c,d
abra**c**adabra	0,1,2,2	a,r,b,**c**,d
abrac**a**dabra	0,1,2,2,3	c,**a**,r,b,d

Move to Front

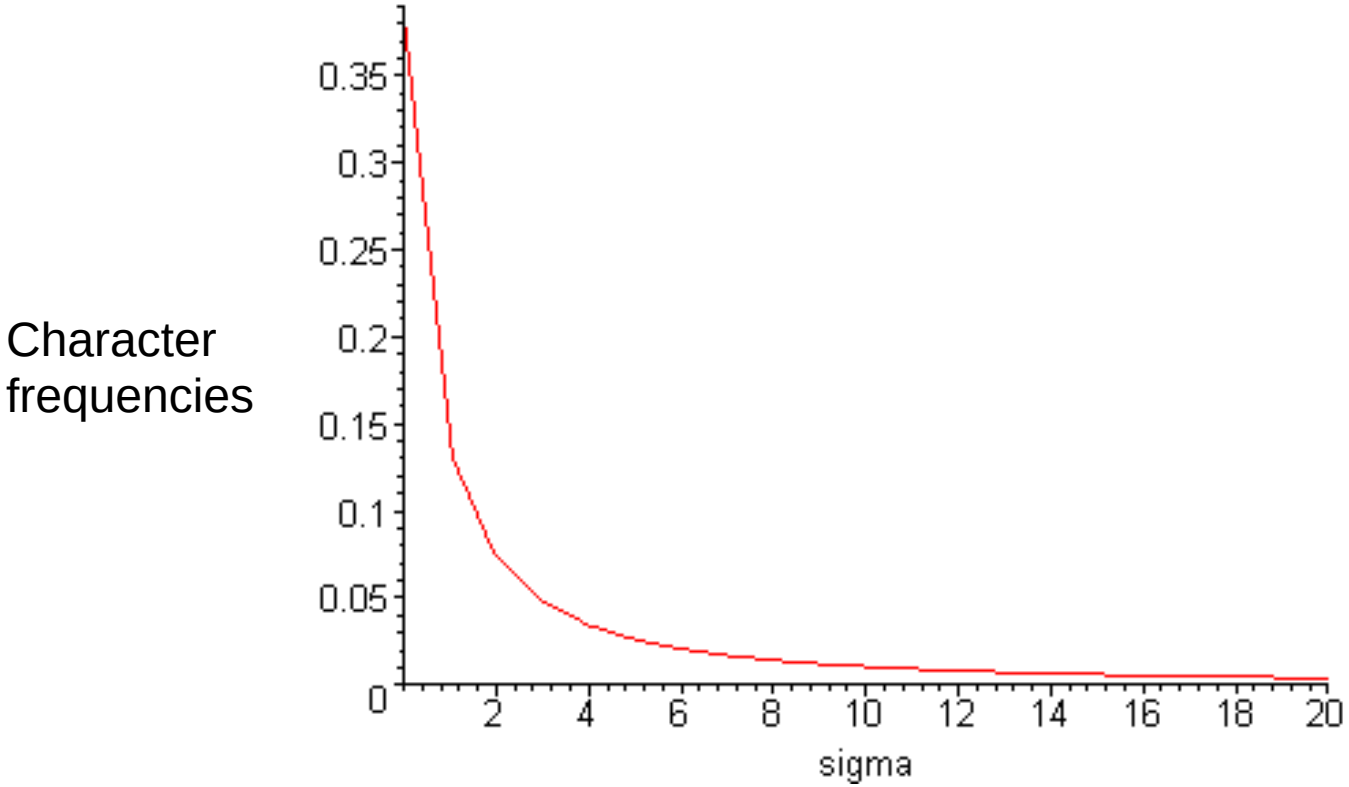
abracadabra		**a**,b,r,c,d
a**b**racadabra	0	a,**b**,r,c,d
ab**r**acadabra	0,1	b,a,**r**,c,d
abr**a**cadabra	0,1,2	r,b,**a**,c,d
abra**c**adabra	0,1,2,2	a,r,b,**c**,d
abrac**a**dabra	0,1,2,2,3	c,**a**,r,b,d
abraca**d**abra	0,1,2,2,3,1	a,c,r,b,**d**

Move to Front

abracadabra		**a**,b,r,c,d
a**b**racadabra	0	a,**b**,r,c,d
ab**r**acadabra	0,1	b,a,**r**,c,d
abr**a**cadabra	0,1,2	r,b,**a**,c,d
abra**c**adabra	0,1,2,2	a,r,b,**c**,d
abrac**a**dabra	0,1,2,2,3	c,**a**,r,b,d
abraca**d**abra	0,1,2,2,3,1	a,c,r,b,**d**
abracadabra	0,1,2,2,3,1,4,1,4,4,2	

After MTF

- Now we have a string with small numbers: lots of 0s, many 1s, …
- Skewed frequencies: Run Arithmetic!



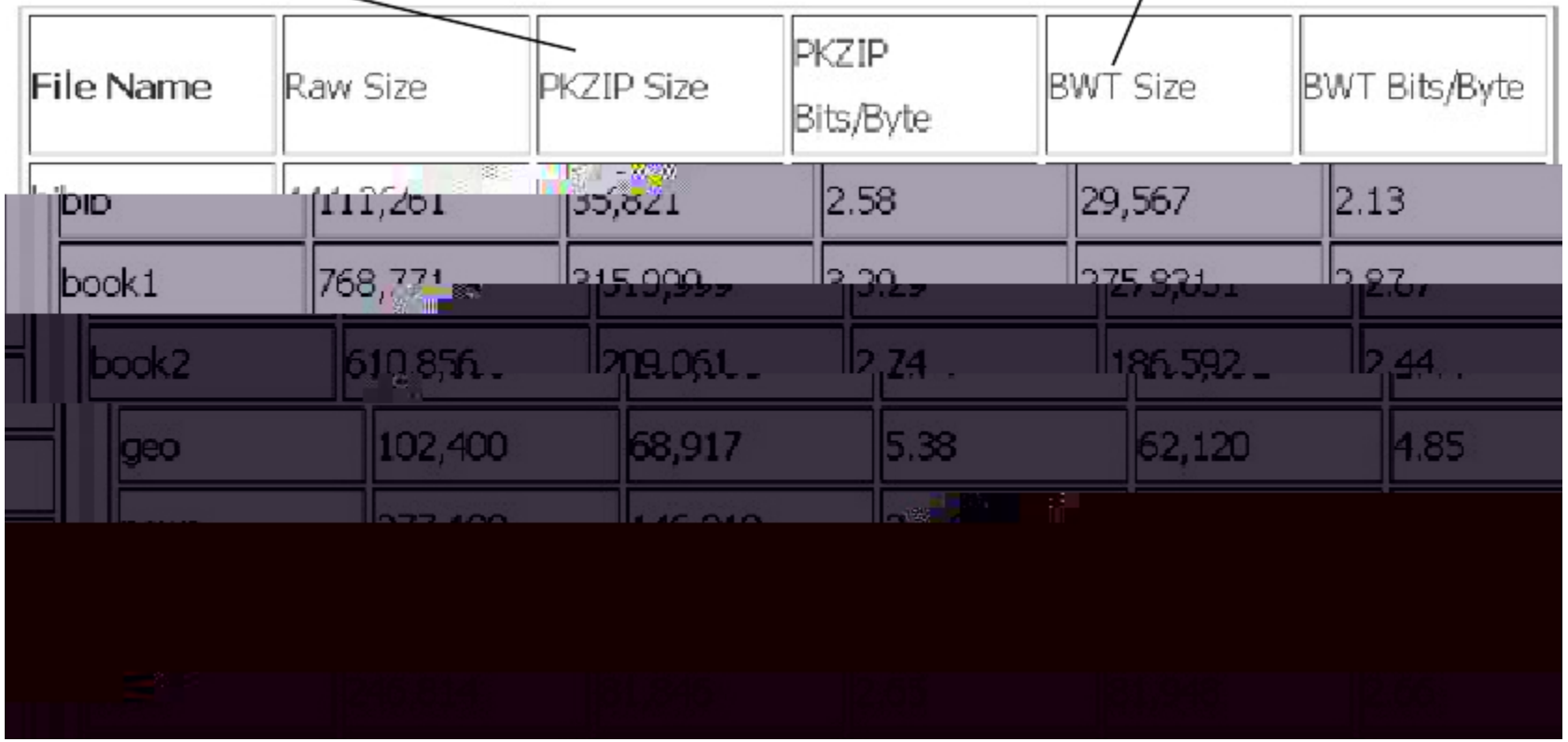
:

	0	..
	1	..
	1	..
	0	..
	1	..
	1	..
	2	..
	1	..
	3	..

, 256

.

(. .,) + + +



File Name	Raw Size	PKZIP Size	PKZIP Bits/Byte	BWT Size	BWT Bits/Byte
bib	111,261	35,821	2.58	29,567	2.13
book1	768,771	315,999	3.29	275,831	2.87
book2	610,856	209,061	2.74	186,592	2.44
geo	102,400	68,917	5.38	62,120	4.85

$= \quad (\ )$

0 -1,

:

0

-1 .

, , ,

.

		#
0		0
1		0
2		
3		
4		
5		
6		
7		2

	0
	3
	4
	6
	7

???????

		#
0		0
1		0
2		
3		
4		
5		
6		
7		2

	0
	3
	4
	6
	7

??????

		#
0		0
1		0
2		
3		
4		
5		
6		
7		2

	0
	3
	4
	6
	7

?????

		#
0		0
1		0
2		
3		
4		
5		
6		
7		2

	0
	3
	4
	6
	7

????

		#
0		0
1		0
2		
3		
4		
5		
6		
7		2

	0
	3
	4
	6
	7

???

		#
0		0
1		0
2		
3		
4		
5		
6		
7		2

	0
	3
	4
	6
	7

??

		#
0		0
1		0
2		
3		
4		
5		
6		
7		2

	0
	3
	4
	6
	7

?

		#
0		0
1		0
2		
3		
4		
5		
6		
7		2

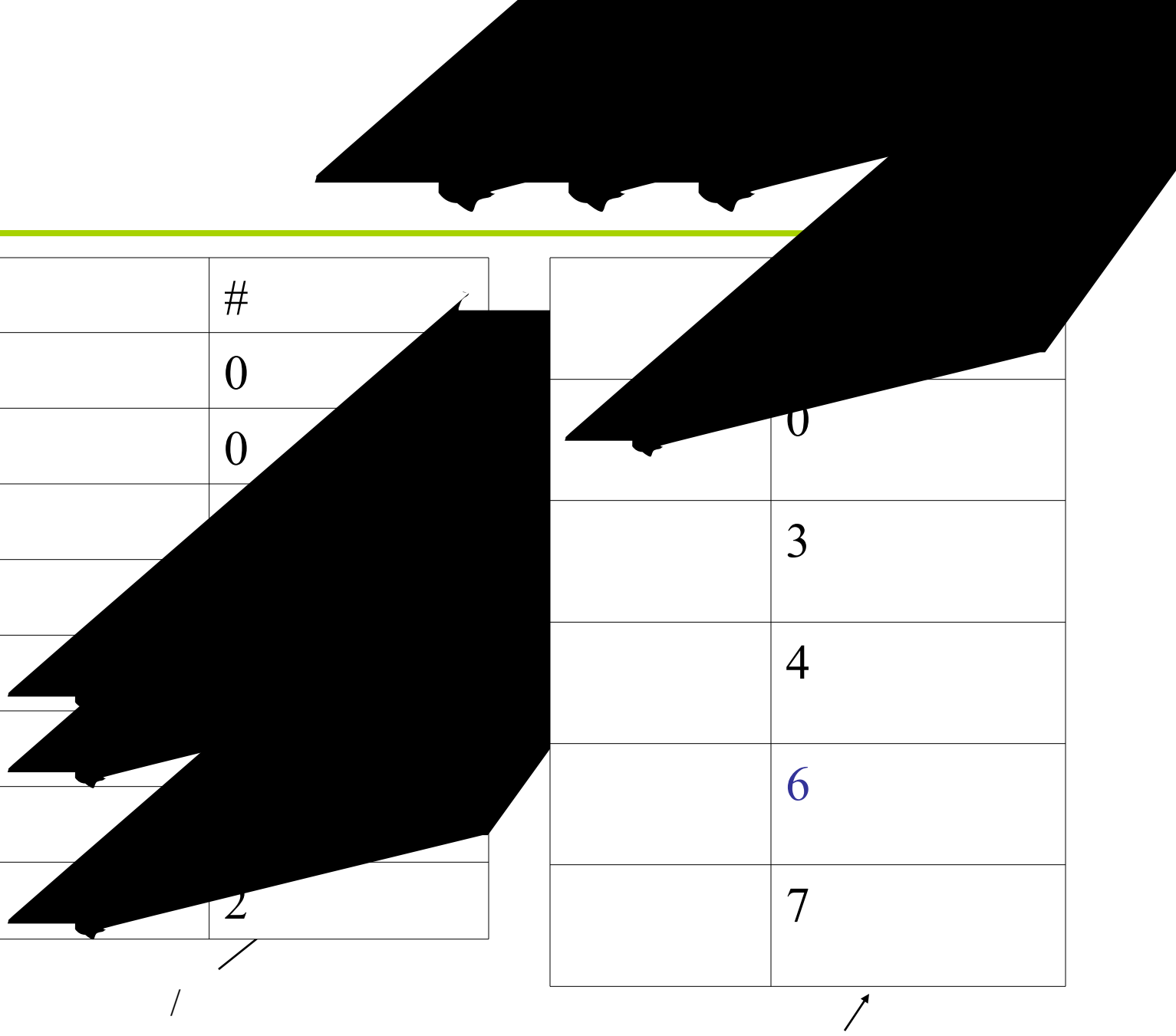
	0
	3
	4
	6
	7

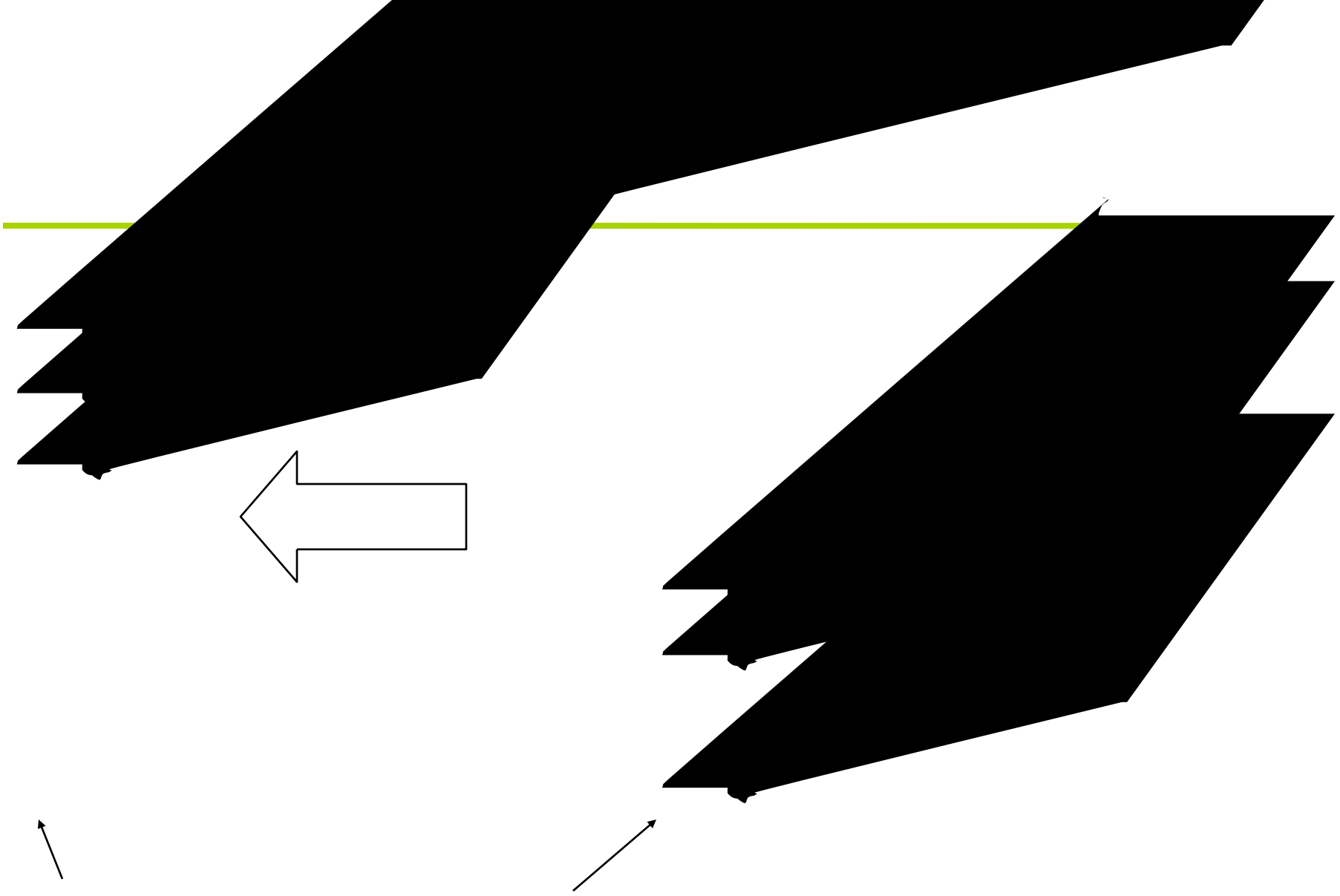
		#
0		0
1		0
2		
3		
4		
5		
6		
7		2

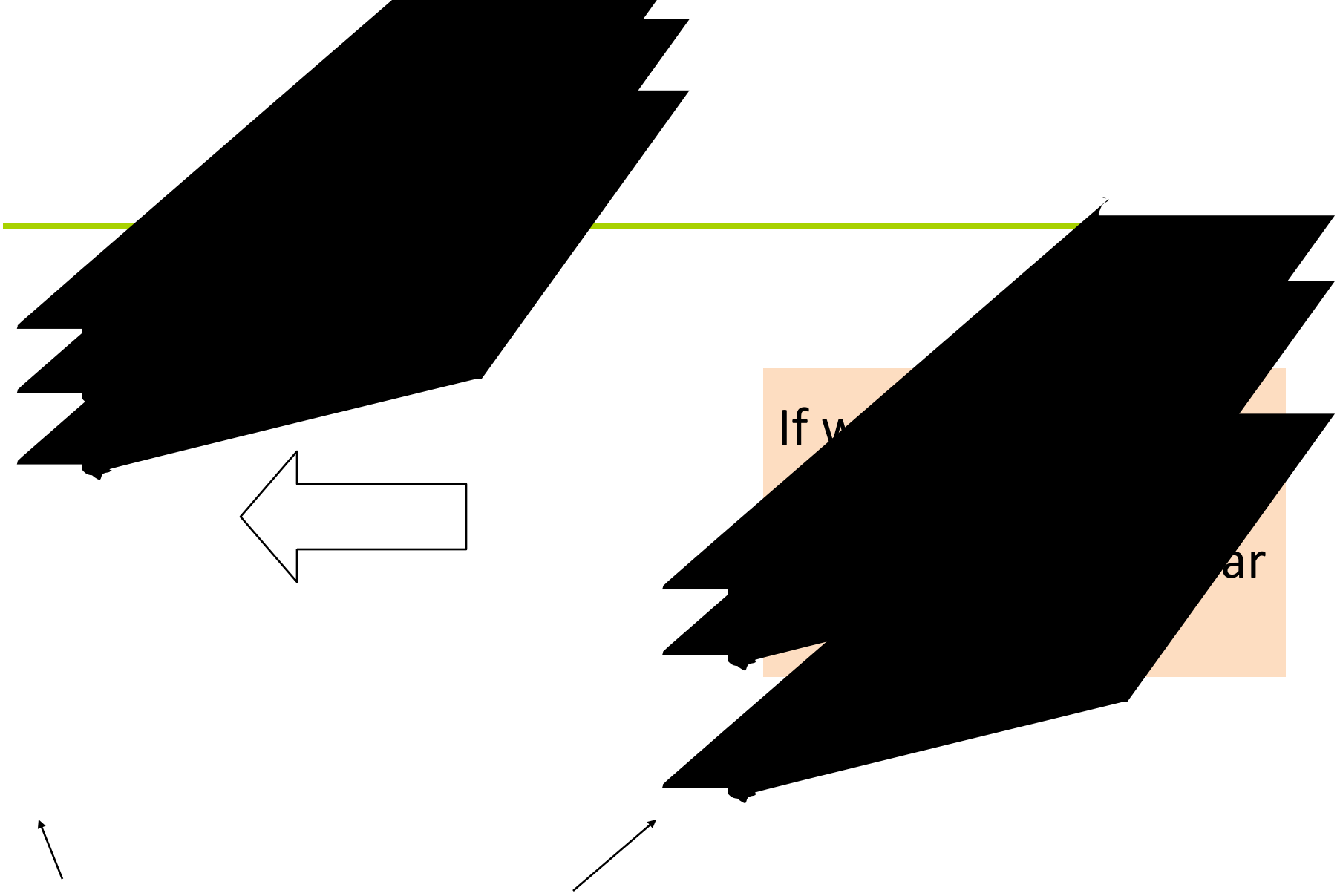
	0
	3
	4
	6
	7

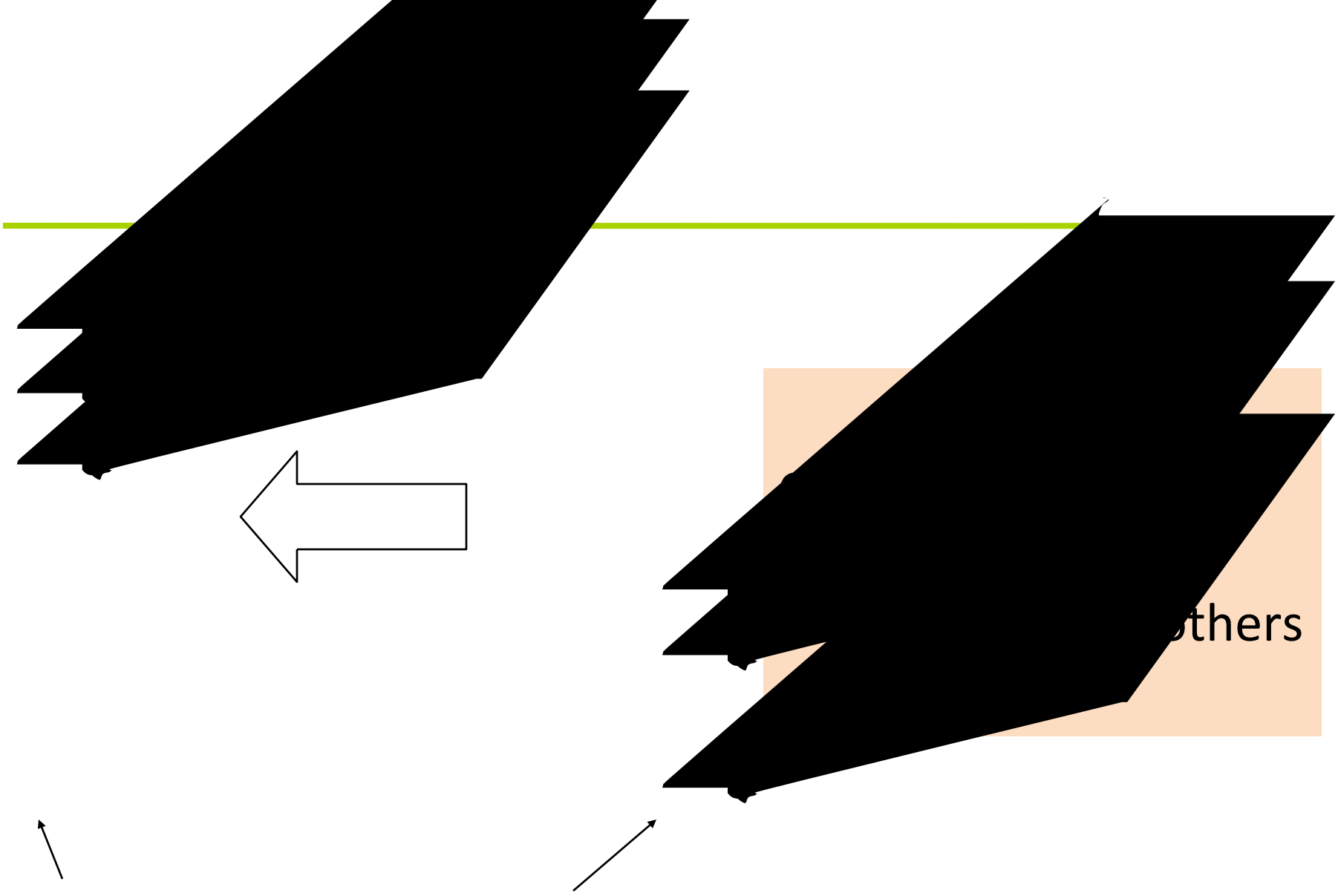
		#
0		0
1		0
2		
3		
4		
5		
6		
7		2

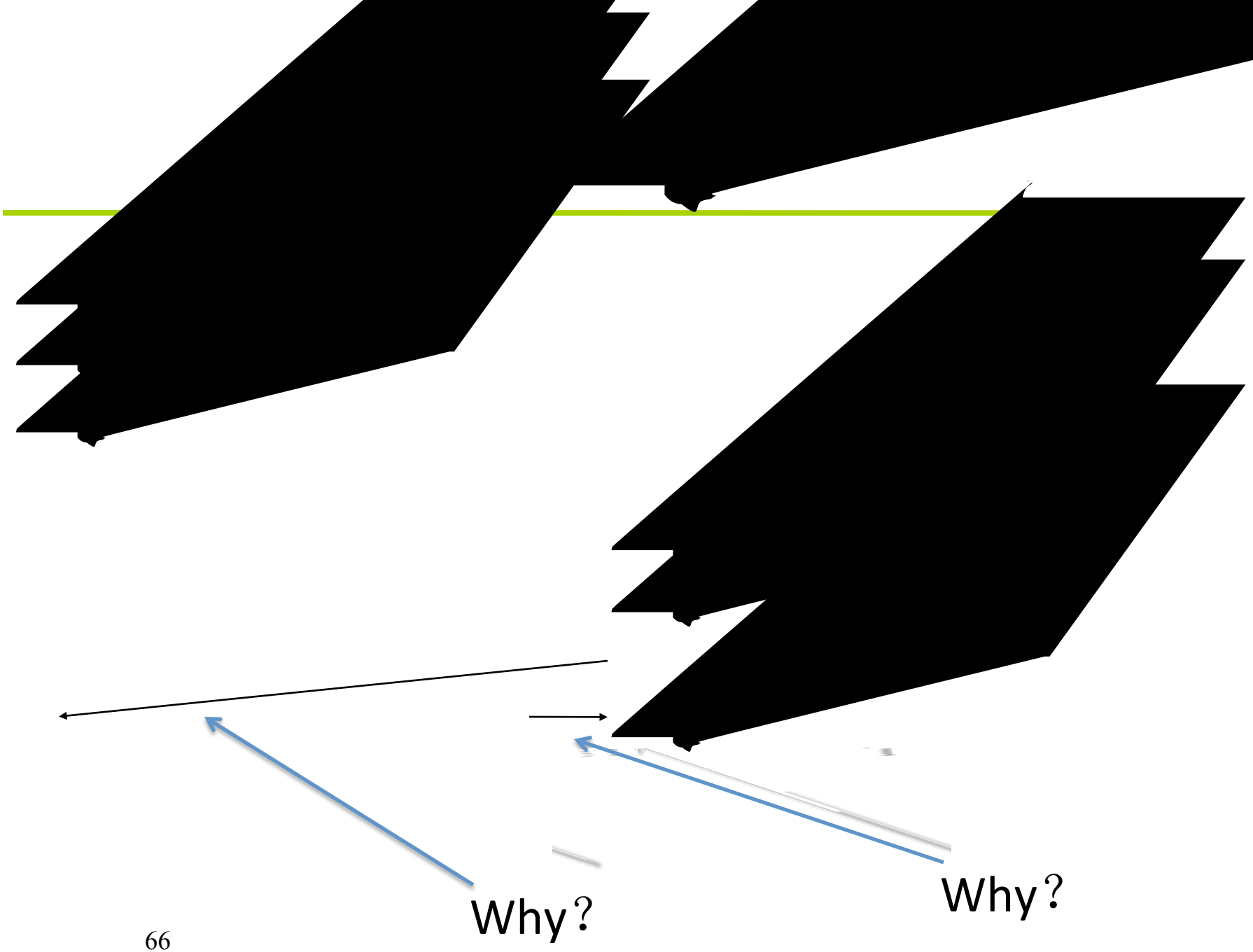
	0
	3
	4
	6
	7

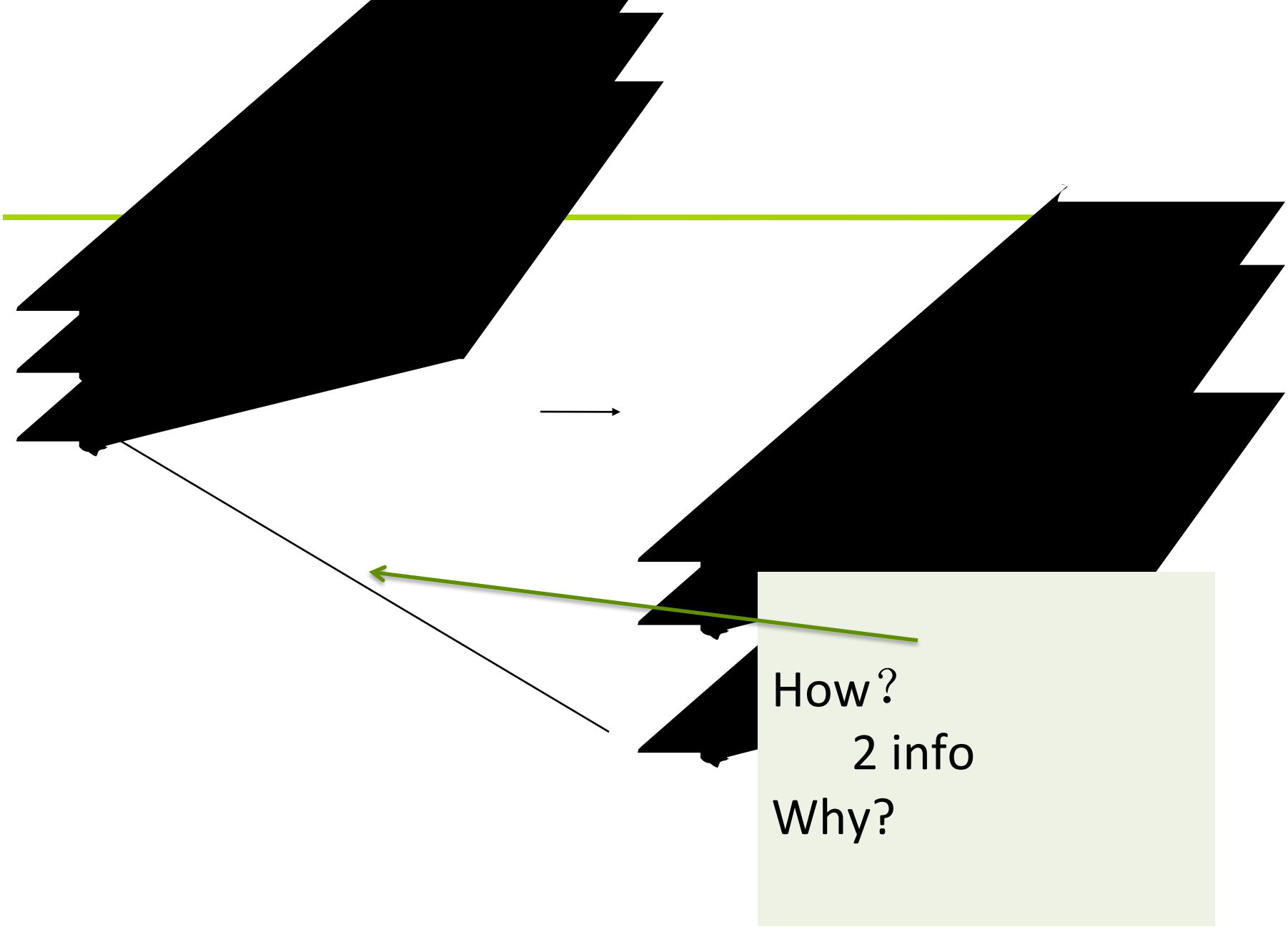


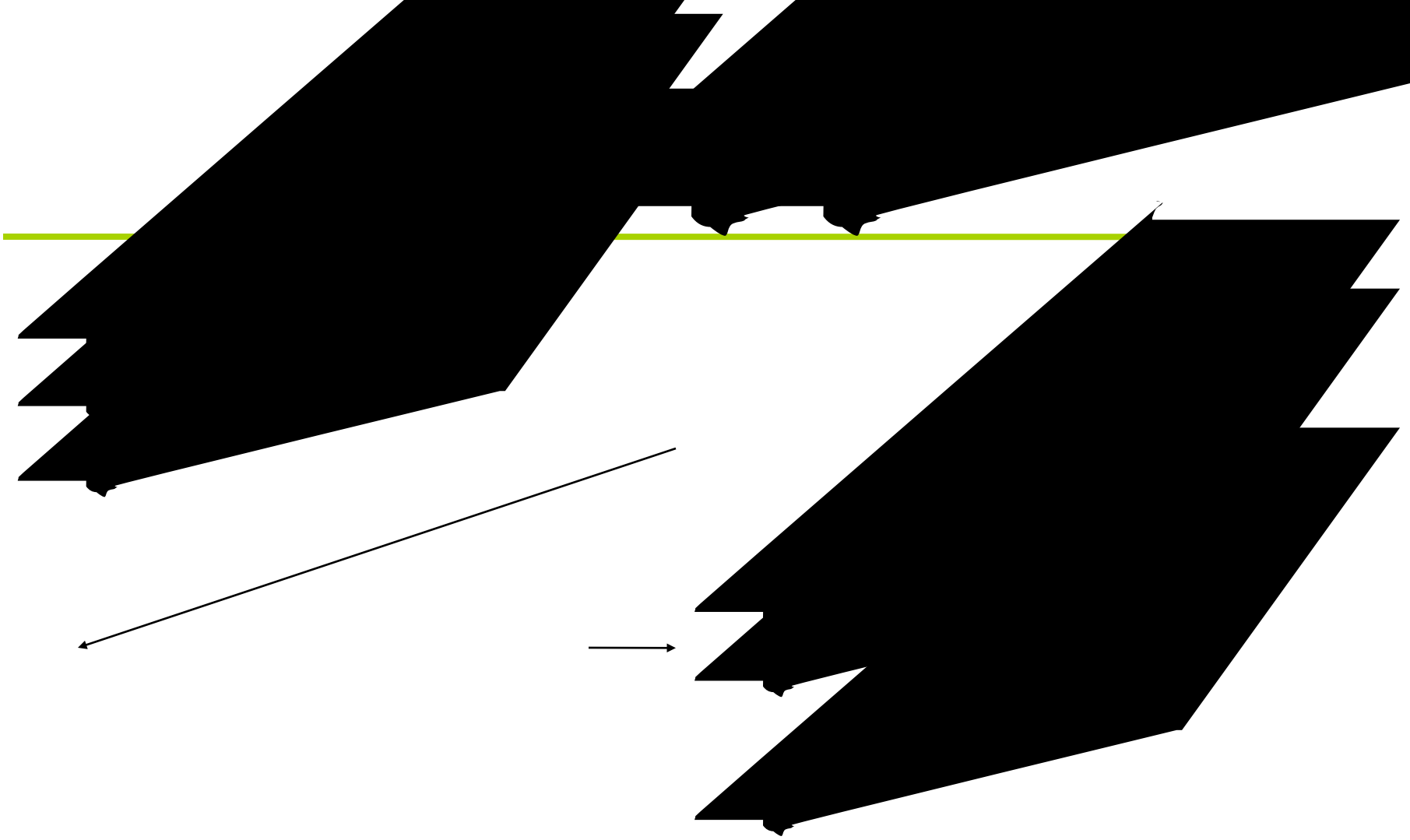


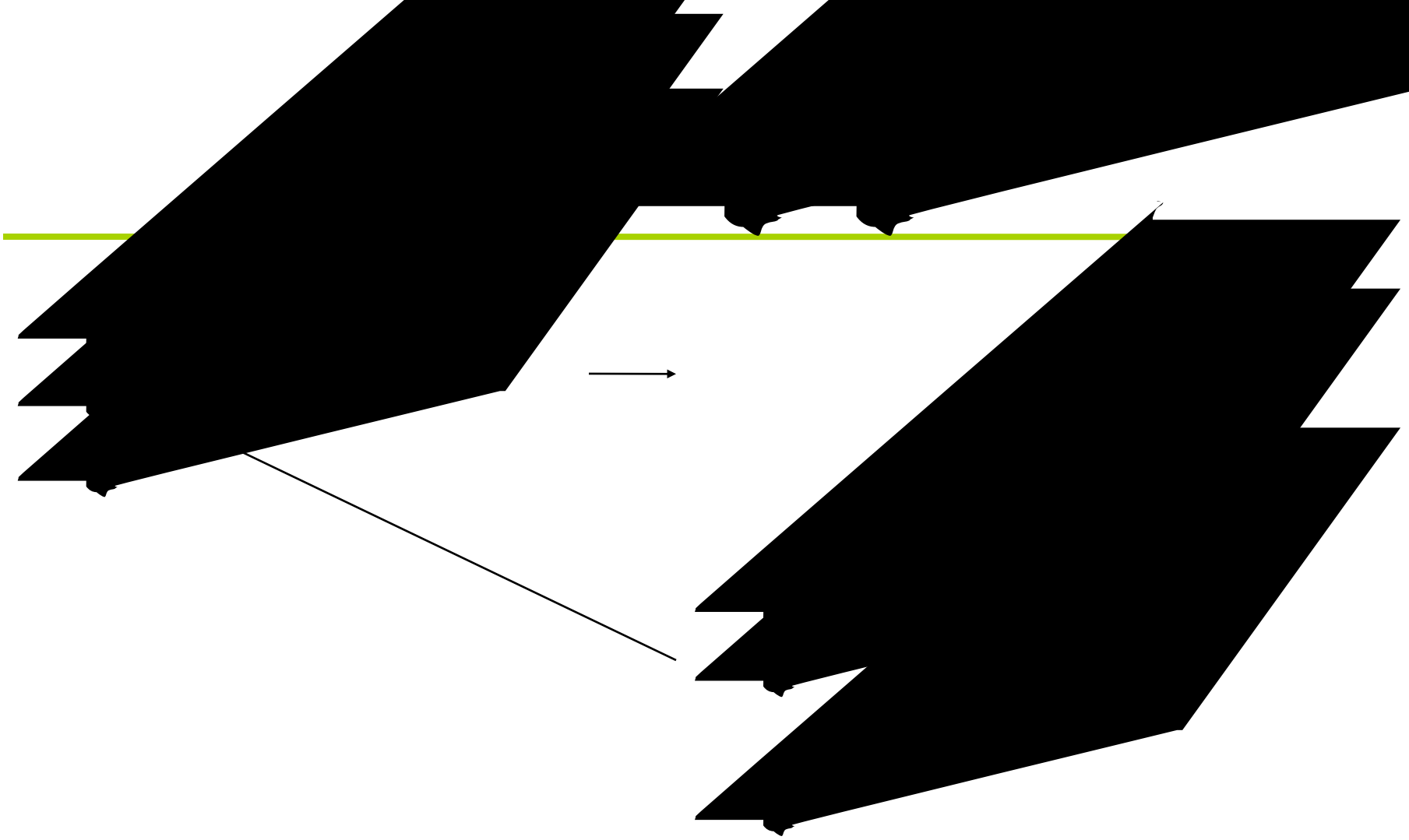

If w
ar

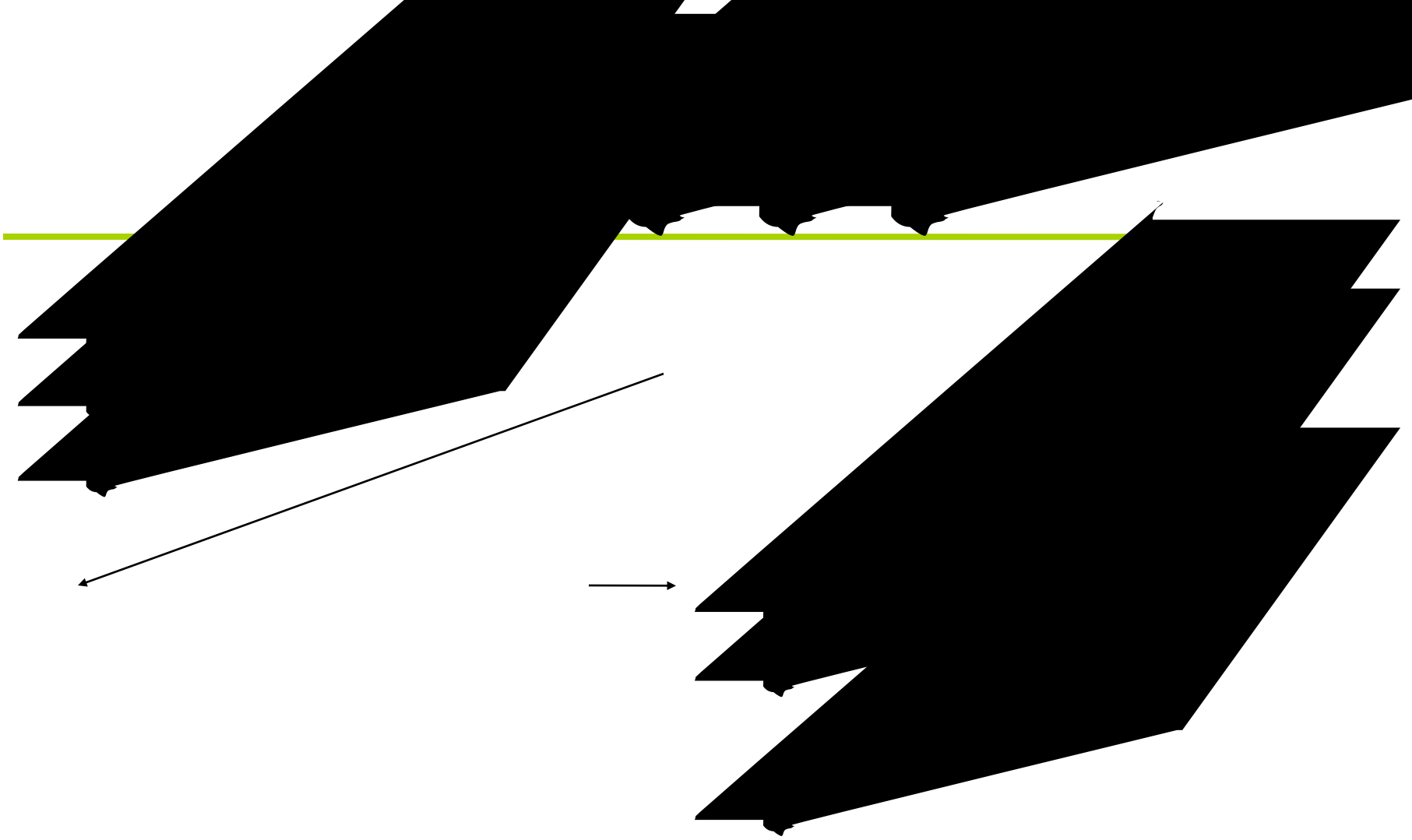

thers

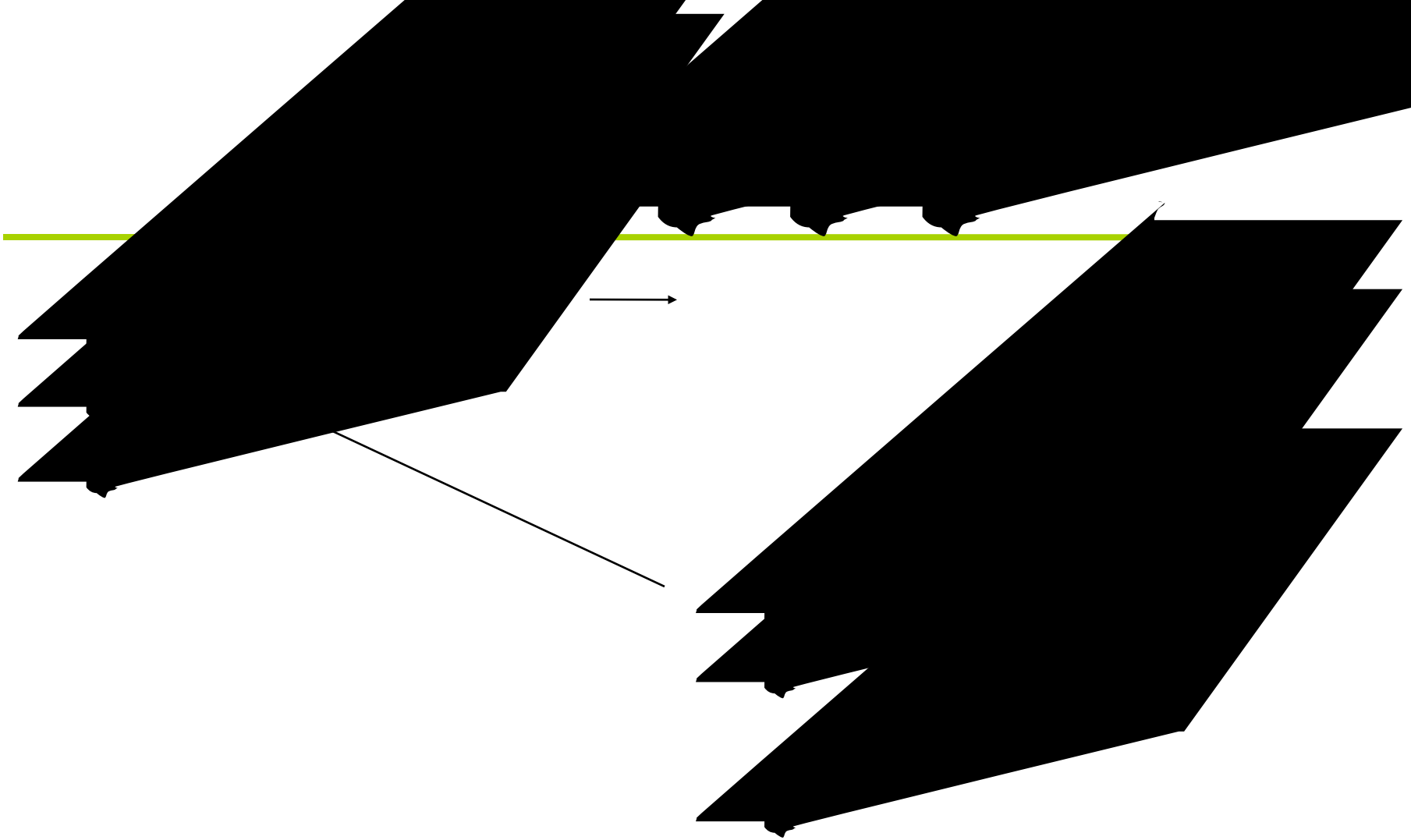

Why?
Why?

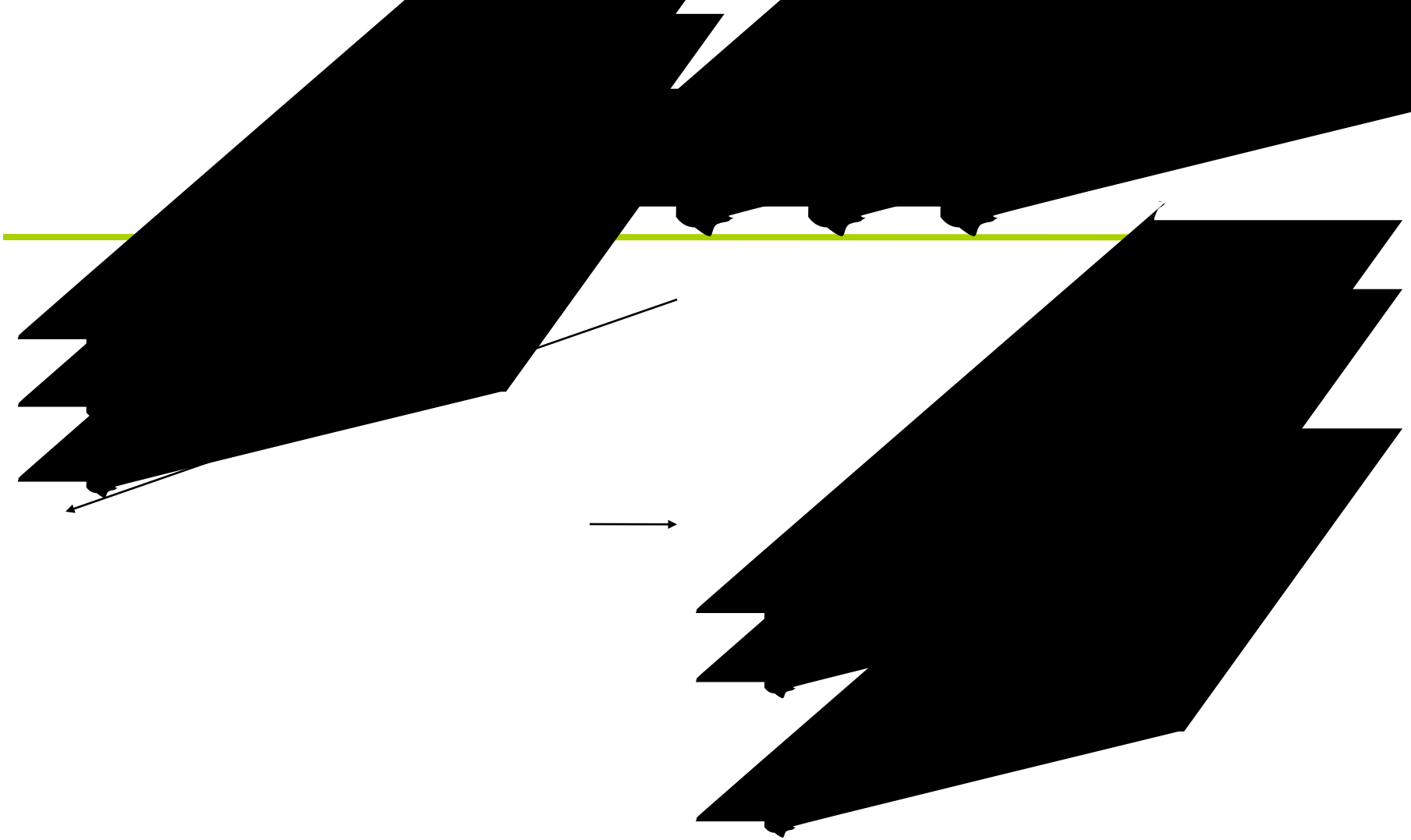

How?
2 info
Why?



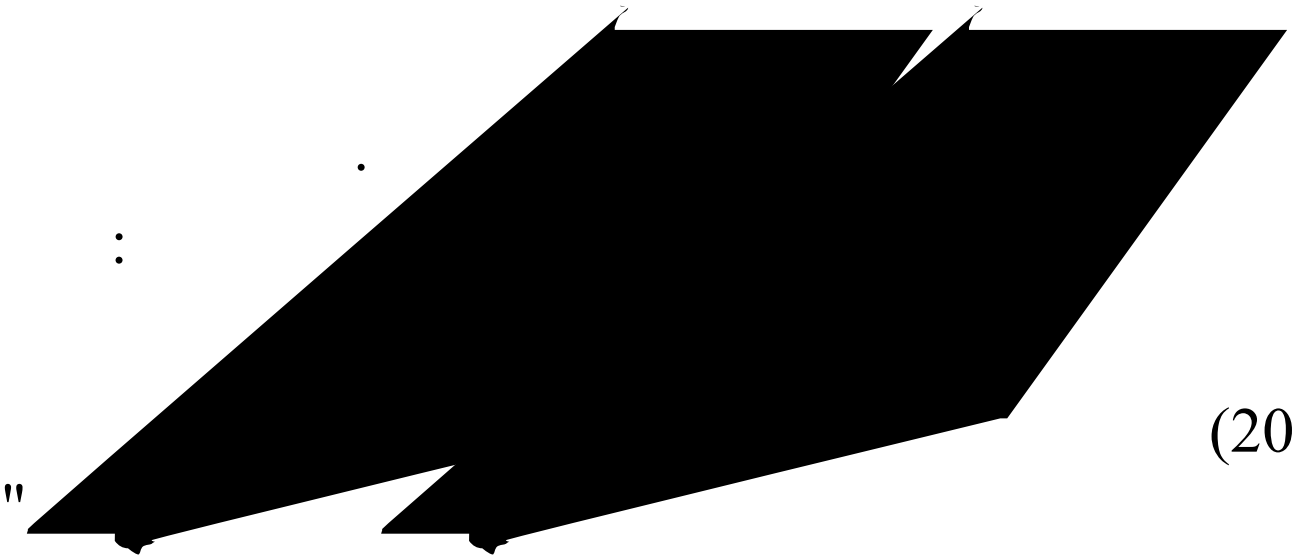








?



(2009).

410

(43): 4350.