



TK1104 – Digital teknologi

Første repetisjonsforelesning

Denne forelesningsøkten vil bli tatt opp og lagt ut i emnet i etterkant.

Hvis du ikke vil være med på opptaket:

	La være å delta med webkameraet ditt.
	La være å delta med mikrofonen din.
To: Marianne Sundby (Privately) Type message here...	Still spørsmål i Chat i stedet for som lyd. Hvis du ønsker kan spørsmålet også sendes privat til foreleser.

Hvem er jeg?
Idun Sollie

- Student i 2-klasse –
Intelligente systemer
- Veileder i:
Digital teknologi
Intro til programmering
Databaser

Kontakt meg på Discord eller:
idun.sollie@kristiania.no

Hva skal vi repetere i dag?

Binære tallsystemet

Toerkomplement

Heksadesimal

ASCII

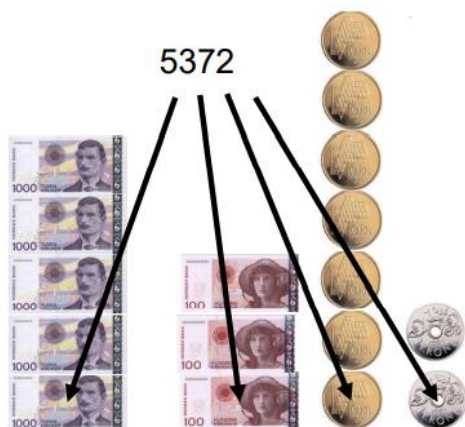
UNICODE og UTF-8

Komprimering

→ LIVE regning

Posisjons talsystem

- Posisjonen avgjør tallets verdi
- Samme som vi bruker til vanlig, med base 10
- Binær talsystem har basen 2
- Brukes for å kunne snakke med datamaskiner



Eksempel med base 6:

$$6^0 = 1$$

$$6^1 = 6$$

$$6^2 = 36$$

$$6^3 = 216$$

$$6^4 = 1\,296$$

$$6^5 = 7\,776$$

$$5302_6 =$$

$$5 \cdot 6^3 + 3 \cdot 6^2 + 0 \cdot 6^1 + 2 \cdot 6^0 =$$

$$5 \cdot 216 + 3 \cdot 36 + 0 \cdot 6 + 2 \cdot 1 =$$

$$1080 + 108 + 0 + 2 = 1190_{10}$$

Binære tallsystemet – «ON eller OFF»

0000 0000 ← Alle er «OFF» = ikke regn med disse

1111 1111 ← Alle er «ON» = regn med disse

0000 0001 ← Første bit er «ON» = regn med denne
 $2^7 \ 2^6 \ 2^5 \ 2^4 \ 2^3 \ 2^2 \ 2^1 \ 2^0$

- $2^0 = 1$
- $0000 \ 0001_2 = 1_{10}$

0001 0100 ← Tredje og femte bit er «ON» = regn med disse
 $2^7 \ 2^6 \ 2^5 \ 2^4 \ 2^3 \ 2^2 \ 2^1 \ 2^0$

- $2^2 = 4$
- $2^4 = 16$
- $4 + 16 = 20$
- $0001 \ 0100_2 = 20_{10}$

$$2^0 = 1$$

$$2^1 = 2$$

$$2^2 = 4$$

$$2^3 = 8$$

$$2^4 = 16$$

$$2^5 = 32$$

$$2^6 = 64$$

$$2^7 = 128$$

$$2^8 = 256$$

$$2^9 = 512$$

$$2^{10} = 1024$$

Binære tallsystemet

$$0001 = 0 + 0 + 0 + 2^0 = 0 + 0 + 0 + 1 = 1$$

$$0010 = 0 + 0 + 2^1 + 0 = 0 + 0 + 2 + 0 = 2$$

$$0011 = 0 + 0 + 2^1 + 2^0 = 0 + 0 + 2 + 1 = 3$$

$$0100 = 0 + 2^2 + 0 + 0 = 0 + 4 + 0 + 0 = 4$$

$$0101 = 0 + 2^2 + 0 + 2^0 = 0 + 4 + 0 + 1 = 5$$

$$0110 = 0 + 2^2 + 2^1 + 0 = 0 + 4 + 2 + 0 = 6$$

$$0111 = 0 + 2^2 + 2^1 + 2^0 = 0 + 4 + 2 + 1 = 7$$

$$1000 = 2^3 + 0 + 0 + 0 = 8 + 0 + 0 + 0 = 8$$

$$1001 = 2^3 + 0 + 0 + 2^0 = 8 + 0 + 0 + 1 = 9$$

$$1010 = 2^3 + 0 + 2^1 + 0 = 8 + 0 + 2 + 0 = 10$$

$$2^0 = 1$$

$$2^1 = 2$$

$$2^2 = 4$$

$$2^3 = 8$$

$$2^4 = 16$$

$$2^5 = 32$$

$$2^6 = 64$$

$$2^7 = 128$$

$$2^8 = 256$$

$$2^9 = 512$$

$$2^{10} = 1024$$

Binære tallsystemet

0101 1100 ← 8 bit presisjon

0000 0001 0101 1100 ← 16 bit presisjon

0101 1100₂ =

64 + 16 + 8 + 4 =

92₁₀

0000 0001 0101 1100₂ =

256 + 64 + 16 + 8 + 4 =

348₁₀

2⁰ = 1

2⁸ = 256

2¹ = 2

2⁹ = 512

2² = 4

2¹⁰ = 1 024

2³ = 8

2¹¹ = 2 048

2⁴ = 16

2¹² = 4 096

2⁵ = 32

2¹³ = 8 192

2⁶ = 64

2¹⁴ = 16 384

2⁷ = 128

2¹⁵ = 32 768

Addisjon med binære tall

Overflow hvis dette er 8 bit presisjon

$$\begin{array}{r}
 \overset{1}{\text{1}} \quad \overset{1}{\text{0}} \overset{1}{\text{1}} \overset{1}{\text{1}} \overset{1}{\text{0}} \overset{1}{\text{0}} \\
 + \quad \text{1100 1110} \\
 \hline
 \text{0010 1010}
 \end{array}$$

Svar i 8 bit presisjon:
0010 1010

Svar i 16 bit presisjon:
0000 0001 0010 1010

$0 + 0 = 0$	$0 + 1 = 1$	$1 + 0 = 1$
$1 + 1 =$ 0 og 1 i mente		$1 + 1 + 1 =$ 1 og 1 i mente

Ser mer på dette under live regning 😊

Subtraksjon med binære tall

The diagram illustrates the two-step process of finding the 2's complement of the binary number 1100 1110:

- Flip og +1:** The original number 1100 1110 is flipped to 0101 1100. Then, 1 is added to the least significant bit (LSB) to get 0101 1101.
- Addisjon:** The flipped number 0101 1100 is added to the result from the first step, 0101 1101, to produce the final 2's complement result: 1000 1110.

Toerkomplement

- Brukes for å kunne representere negative tall
- Fungerer ved bestemt presisjon feks 8 eller 16 bit
- Vi kommer se på toerkomplement med 8 bit
- Mest signifikante bit, MSB, indikerer om det er et negativt tall eller ikke

MSB
↙
0111 1111 ← Største positive tallet i 8 bit presisjon er +127

1000 0000 ← Den laveste tallet i 8 bit presisjon er -128

1111 1111 ← $-128 + (64 + 32 + 16 + 8 + 4 + 2 + 1) = -128 + 127 = -1$

Toerkomplement – binært til desimalt

Hva vet vi?

- 8 bit presisjon
- Vi snakker om toerkomplement

$$\mathbf{0001\ 0101} =$$

$$\mathbf{0} + (16 + 4 + 1) =$$

$$\mathbf{0} + 21 =$$

$$\mathbf{21}$$

$$\mathbf{1001\ 0101} =$$

$$\mathbf{-128} + (16 + 4 + 1) =$$

$$\mathbf{-128} + 21 =$$

$$\mathbf{-107}$$

Toerkomplement – desimalt til binært

Hva vet vi?

- 8 bit presisjon
- Vi snakker om toerkomplement

0	0	0	1	0	1	0	1
128	64	32	16	8	4	2	1

$2^0 = 1$

$2^1 = 2$

$2^2 = 4$

$2^3 = 8$

$2^4 = 16$

$2^5 = 32$

$2^6 = 64$

$2^7 = 128$

$2^8 = 256$

$2^9 = 512$

$2^{10} = 1024$

$$\left. \begin{array}{l} 21 = \\ 21 - 16 = \\ 5 - 4 = \\ 1 \end{array} \right\} \text{Svar: } 21_{10} = 0001\ 0101_2$$

Toerkomplement – desimalt til binært

Hva vet vi?

- 8 bit presisjon
- Vi snakker om toerkomplement

-107

↓ Konverter +107 til binært

$$107 - 64 =$$

$$43 - 32 =$$

$$11 - 8 =$$

$$3 - 2 = 1$$

0 **1** **1** **0** **1** **0** **1** **1**
 128 64 32 16 8 4 2 1

↓ Flip og +1

1001 0100
 + 1
 —————

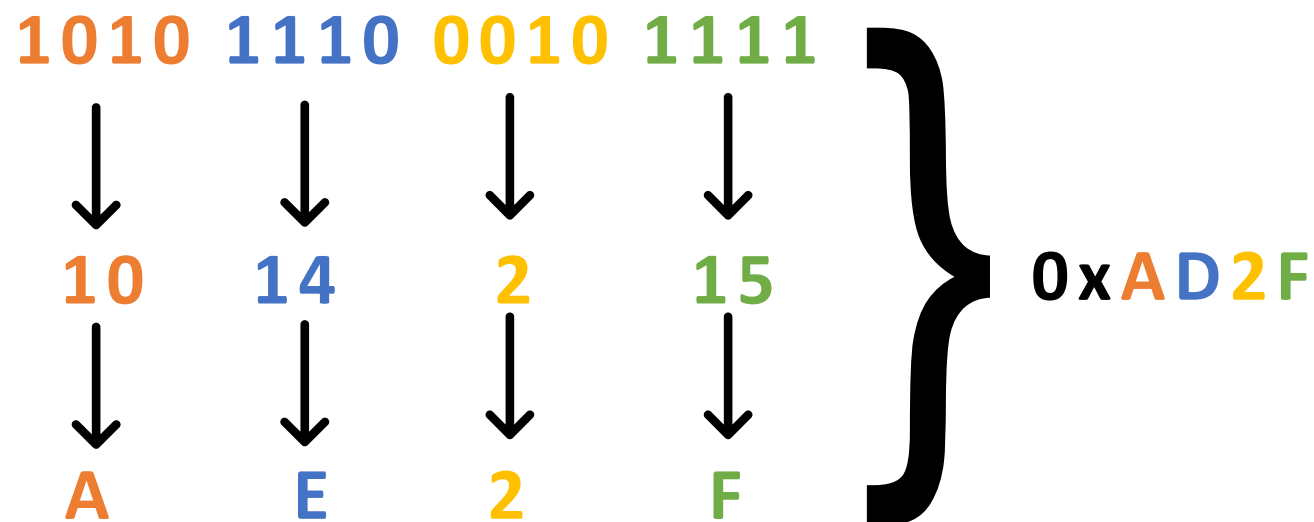
1001 0101

↓ Sjekker om det er riktig

$$\begin{aligned}
 & -128 + (16 + 4 + 1) = \\
 & -128 + 21 = -107
 \end{aligned}$$

Heksadesimal

- Tallsystem med 16 som base
- Bokstavene **A-F** er en verdi av et tall
- Brukes for å spare plass og mulighet for å skrive binære tall mer kompakt
- **0x** indikerer at det er et heksadesimal



A = 10

B = 11

C = 12

D = 13

E = 14

F = 15

Heksadesimal

$$0x002F + 0x0050 = ?$$



Konverter til binært

$$\begin{array}{r} 0000\ 0000\ 0010\ 1111 \\ + 0000\ 0000\ 0101\ 0000 \\ \hline 0000\ 0000\ 0111\ 1111 \end{array}$$



Konverter til heksadesimal

Svar: $0x007F$

A = 10

B = 11

C = 12

D = 13

E = 14

F = 15

ASCII – American Standard Code

- 7 bit, siste biten er paritetsbit
- Tegnkoding for datamaskiner
- Fra binært → Glyf
- Blev utvidet med feks. ISO 8859-1 og Windows 1252
- ASCII tegn har samme verdi i UNICODE

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	 	Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	\$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	((72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051))	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[[123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174	|	
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135]]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177		DEL

Source: www.LookupTables.com

UNICODE 🤔

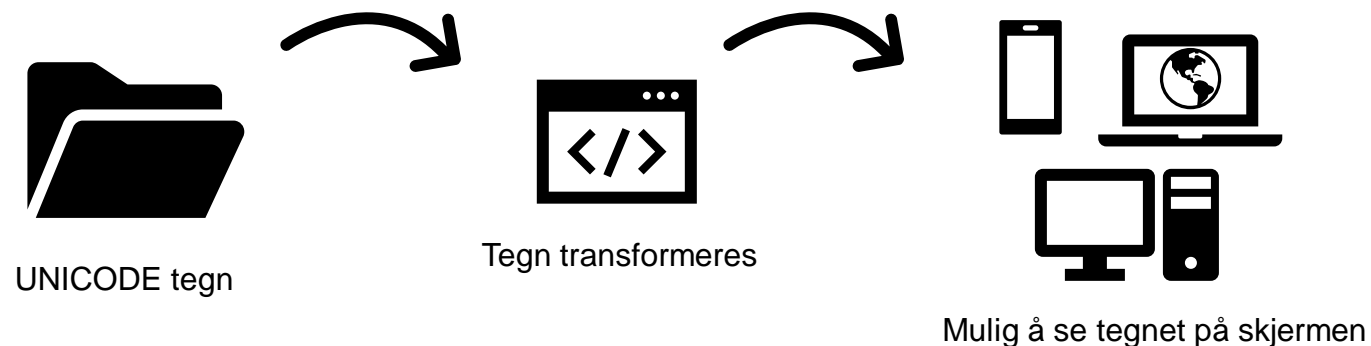
- Delt opp i 17 plan
- Kodepunkter
- Mulig å sende informasjon mellom alle, uavhengig av språk
- Alle typer av tegn og emojiis
- Ulike transformasjonsformat, feks. UTF-16 eller UTF-8
- Transformasjonformatet sier hvordan tegnene skal kodes binært

 U+1F499	 U+203D	 U+FF0D	 U+1D54	 U+266B	 U+30C9
 U+0DC2	 U+0918	 U+0923	 U+058A	 U+0FD1	 U+262E

<https://www.youtube.com/watch?v=-n2nIPHEMG8&feature=youtu.be>

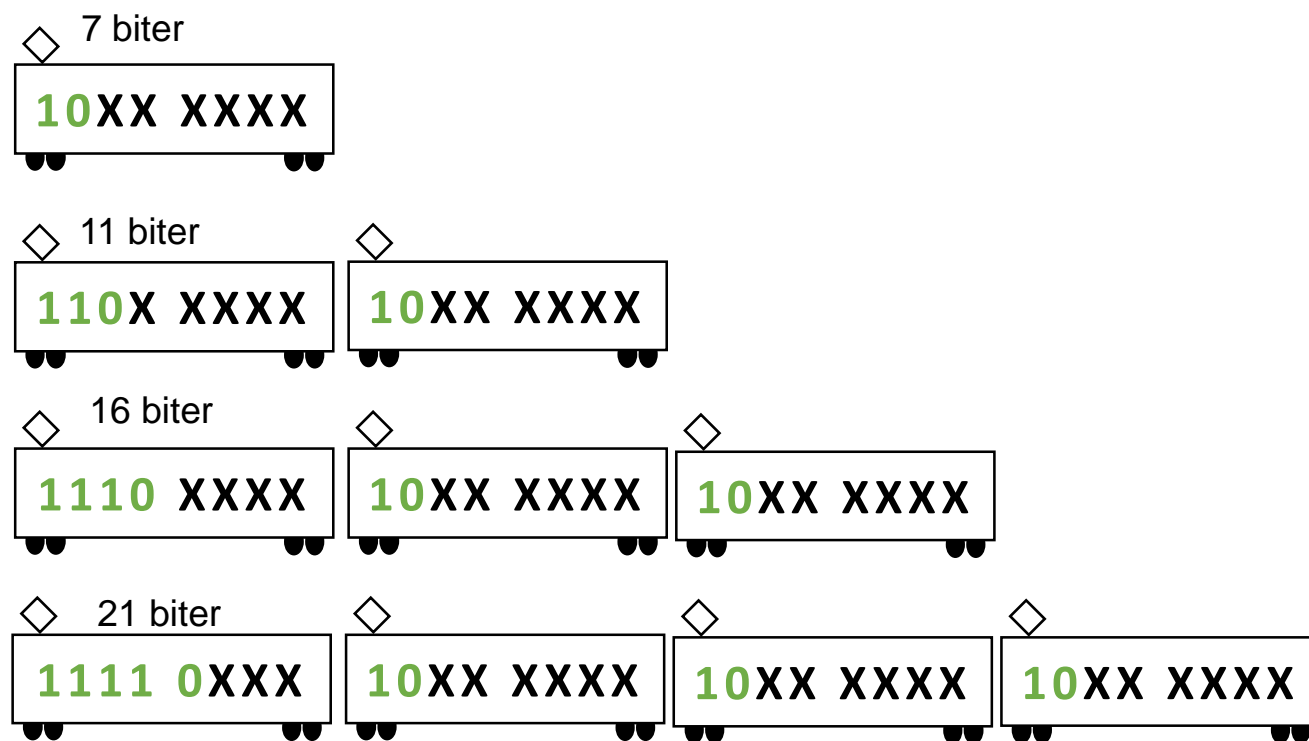
UTF-8 & UTF-16

- Forteller hvordan tegnene fra UNICODE skal kodes binært
- De to mest brukte transformasjonsformatene
- UTF-8 → minimum en byte
- UTF-16 → minimum to byte



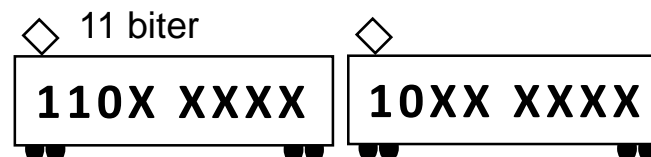
UTF-8: Fra UNICODE kode-punkt til binært med UTF-8

- 'ë' har kode-punkt U+00EB i UNICODE
- Hva er UTF-8 kodningen av U+00EB?
- 8 bit kode-punkter hvilket betyr at vi må bruke 11 biter og 2 byte for å finne svaret



UTF-8

- Hva er UTF-8 kodningen av U+00EB?

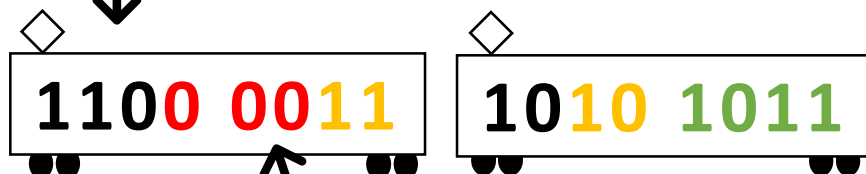


U+00EB

↓ Konverter til binært

0000 0000 1110 1011

↓ Sett in i 11 biter «vognene»



Padding

Konverter svaret fra «vognene» til heksadesimal

1100 0011 1010 1011



C



3



A



B

Svar: UTF-8 kodningen av
U+00EB er C3AB

UTF-8: Adresselokasjon mellom to kode-punkter

Hvor mange adresser er det mellom U+06B4 og U+06FF?

$$(U+06FF - U+06B4) + 1 = ?$$

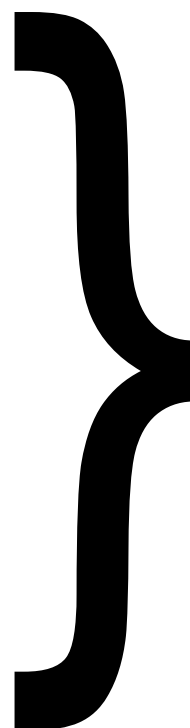
$$U+06FF - U+06B4$$

↓ Konverter til binært

$$\begin{array}{r} 0000\ 0110\ 1111\ 1111 \\ - 0000\ 0110\ 1011\ 0100 \\ \hline \end{array}$$

↓ Flip og +1 (Toerkomplement)

$$\begin{array}{r} 0000\ 0110\ 1111\ 1111 \\ + 1111\ 1001\ 0100\ 1100 \\ \hline 0000\ 0000\ 0100\ 1001 \end{array}$$



↓ + 1 på svaret

$$0000\ 0000\ 0100\ 1010$$

↓ Konverter svaret til heksadesimal
(Hvis oppgaven sier det)

$$0000\ 0000\ 0100\ 1010$$

$$\begin{array}{cccc} \downarrow & \downarrow & \downarrow & \downarrow \\ 0 & 0 & 4 & A \end{array}$$

Svar: 0x004A eller 76₁₀

Komprimering

- Brukes til lyd, bilder etc.
- Vi ser på ordbok-koding
- Lager ordboken i løpet av kodningen

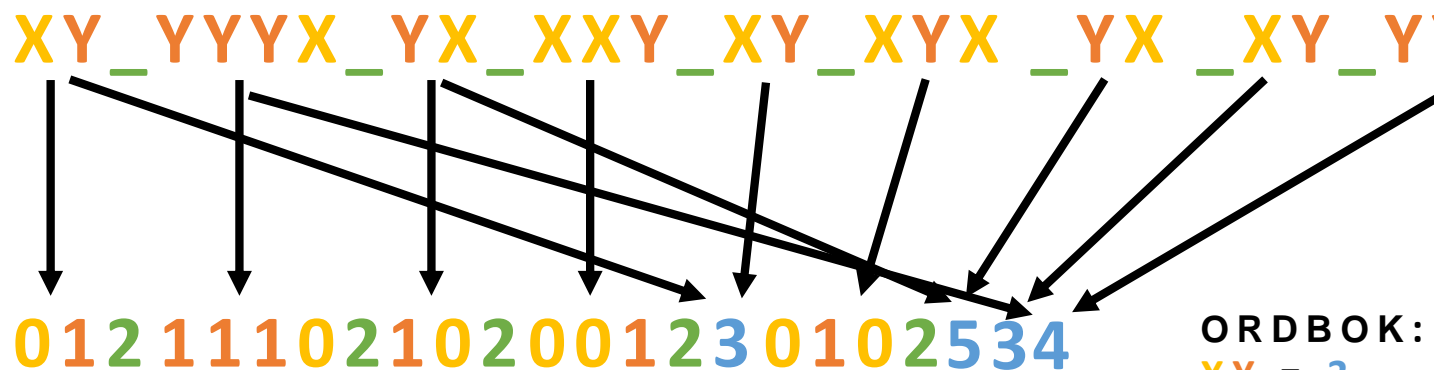
X = 0

Y = 1

_ = 2

XY _ YYYX _ YX _ XXY _ XY _ XYX _ YX _ XY _ YYYX

0 1 2 1 1 1 0 2 1 0 2 0 0 1 2 3 0 1 0 2 5 3 4



ORDBOK:

XY = 3

YYYX = 4

YX = 5

Før: 32 tegn
Etter: 23 tegn

LIVE beregning

