Úsporné DŠ

kuko

11.12.2018

Vybrané partie z dátových štruktúr

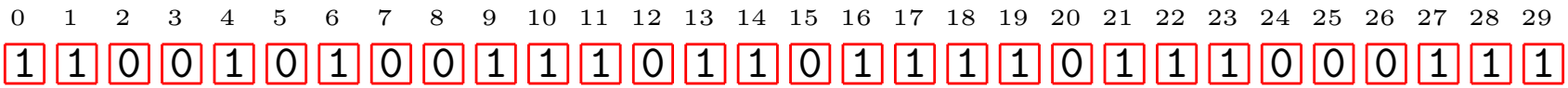
- kompaktná DŠ zaberá $O(OPT)$ bitov
- úsporná DŠ zaberá $OPT + o(OPT)$ bitov
- implicitná DŠ zaberá $OPT + O(1)$ bitov

Rank a Select

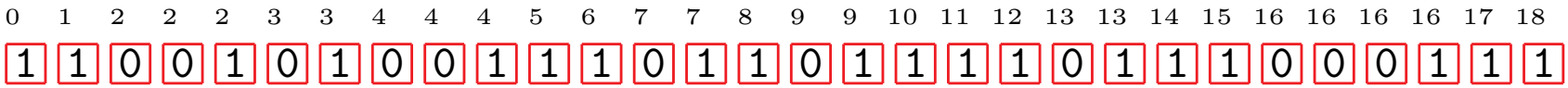
Rank a Select

- Dané: string S
- Query:
  - $\text{rank}_c(S, i) = \#$znakov c od začiatku po i-tu pozíciu
  - $\text{select}_c(S, j) =$pozícia j-teho znaku c v S

Rank

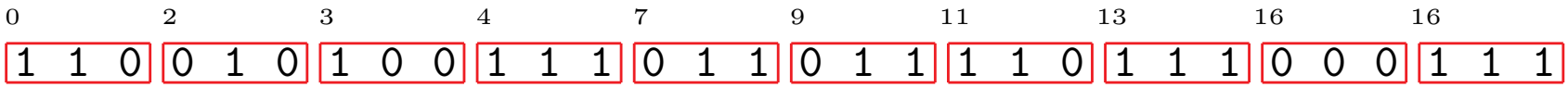

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29
1 1 0 0 1 0 1 0 0 1 1 1 0 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 1

Rank



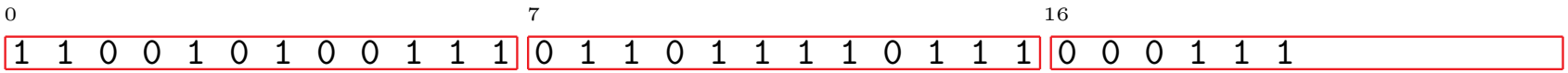
Obr.: predpočítané ranky – $n \lg n$ bitov, $O(1)$ rank

Rank



Obr.: nápad #1: rank iba každých t prvkov

Rank

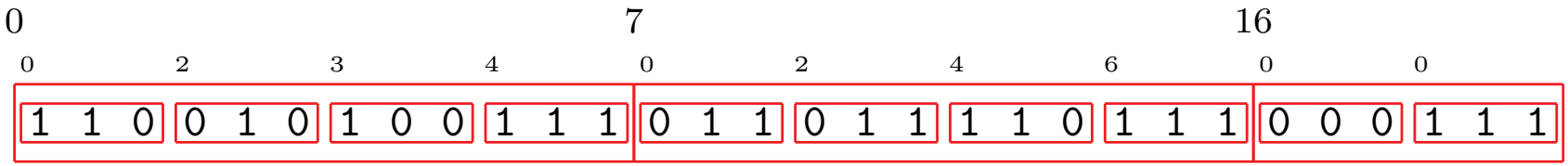


Obr.: väčšie $t \rightarrow$ menej pamäte, viac času

Rank

- jednoduché riešenia:
  - predp. utriedené pozície 1 – $m \lg n$ bitov, $O(1)$ select, $O(\log m)$ rank
  - ranky pre blok dĺžky t: $n + (n/t) \lg n$ bitov, čas $O(t)$

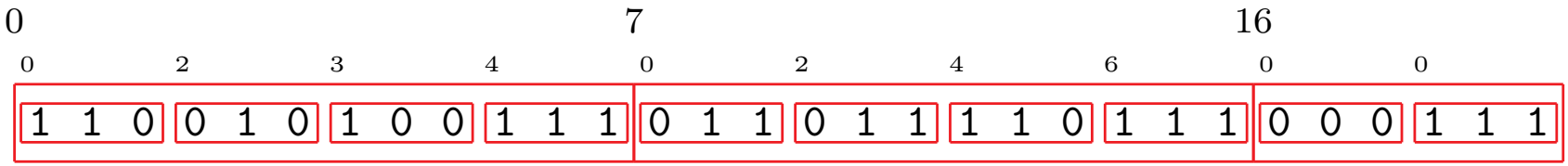
Rank



Obr.: nápad #2: 2 úrovne – superbloky (t_1) a bloky (t_2)

- finta: rank bloku počítame len od začiatku superbloku – stačí menej pamäte
- pre $t_1 = \lg^2 n$ a $t_2 = \frac{1}{2} \lg n$:
  - ranky pre superbloky: $O(\frac{n}{\lg^2 n} \cdot \lg n) = o(n)$ bitov
  - ranky pre bloky: $O(\frac{n}{\lg n} \cdot \lg \lg n) = o(n)$ bitov

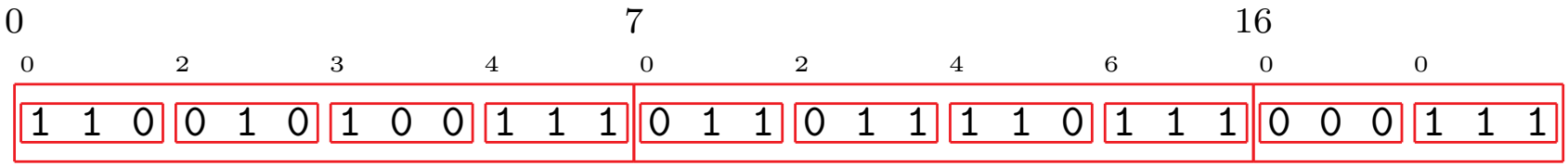
Rank



Obr.: nápad #2: 2 úrovne – superbloky (t_1) a bloky (t_2)

- finta: rank bloku počítame len od začiatku superbloku – stačí menej pamäte
- pre $t_1 = \lg^2 n$ a $t_2 = \frac{1}{2} \lg n$:
  - ranky pre superbloky: $O(\frac{n}{\lg^2 n} \cdot \lg n) = o(n)$ bitov
  - ranky pre bloky: $O(\frac{n}{\lg n} \cdot \lg \lg n) = o(n)$ bitov

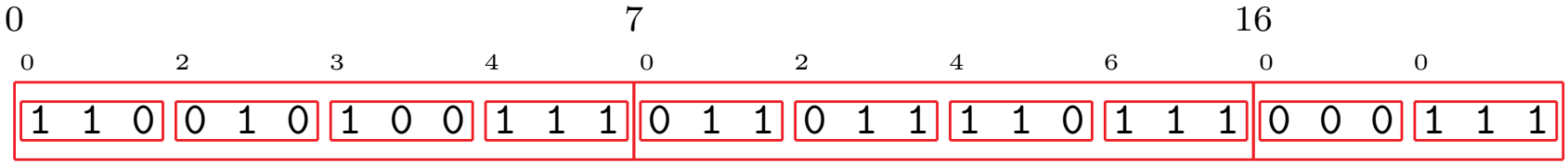
Rank



Obr.: nápad #2: 2 úrovne – superbloky (t_1) a bloky (t_2)

- finta: rank bloku počítame len od začiatku superbloku – stačí menej pamäte
- pre $t_1 = \lg^2 n$ a $t_2 = \frac{1}{2} \lg n$:
  - ranky pre superbloky: $O(\frac{n}{\lg^2 n} \cdot \lg n) = o(n)$ bitov
  - ranky pre bloky: $O(\frac{n}{\lg n} \cdot \lg \lg n) = o(n)$ bitov

Rank



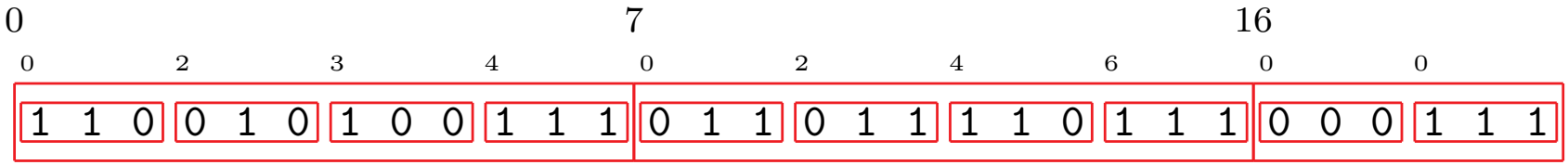
Obr.: nápad #2: 2 úrovne – superbloky (t_1) a bloky (t_2)

- finta: rank bloku počítame len od začiatku superbloku – stačí menej pamäte
- pre $t_1 = \lg^2 n$ a $t_2 = \frac{1}{2} \lg n$:
  - ranky pre superbloky: $O(\frac{n}{\lg^2 n} \cdot \lg n) = o(n)$ bitov
  - ranky pre bloky: $O(\frac{n}{\lg n} \cdot \lg \lg n) = o(n)$ bitov

Rank

Ako spočítame rank v rámci $\frac{1}{2} \lg n$ bitového bloku?

Rank



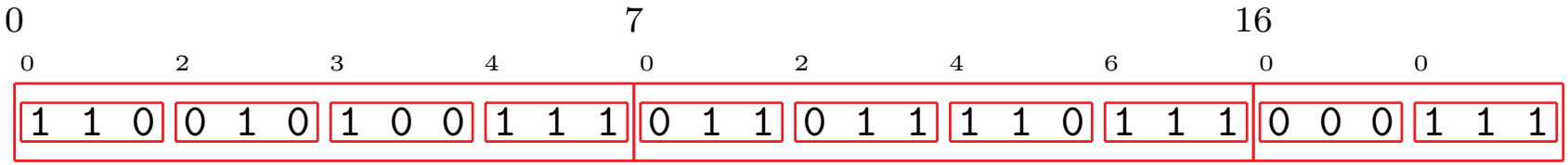
	0	1	2
000	0	0	0
001	0	0	1
010	0	1	1
011	0	1	2
100	1	1	1
101	1	1	2
110	1	2	2
111	1	2	3

Obr.: tabuľka – pre všetky bitstringy a všetky pozície

bitstringov dĺžky $\frac{1}{2}\lg n$ je len $O(\sqrt{n})$

$\Longrightarrow$ pamäť $O(\sqrt{n} \cdot \lg n \cdot \lg\lg n) = o(n)$

Rank



	0	1	2
000	0	0	0
001	0	0	1
010	0	1	1
011	0	1	2
100	1	1	1
101	1	1	2
110	1	2	2
111	1	2	3

Obr.: tabuľka – pre všetky bitstringy a všetky pozície

bitstringov dĺžky $\frac{1}{2}\lg n$ je len $O(\sqrt{n})$
$\Longrightarrow$ pamäť $O(\sqrt{n} \cdot \lg n \cdot \lg\lg n) = o(n)$

Rank

DŠ:

- pôvodný bitstring S
- R_1 – ranky pre superbloky
- R_2 – ranky pre bloky
- R_3 – tabuľka rankov pre krátke bitstringy

$$\text{rank}_1(S, i) = R_1[i/t_1] + R_2[i/t_2] + R_3[S[i' \ldots i' + t_2], i \bmod t_2]$$

- kde $i' \leftarrow i - i \bmod t_2$ (začiatok bloku)

$$\text{rank}_0(S, i) = i - \text{rank}_1(S, i)$$

Rank

DŠ:

- pôvodný bitstring S
- R_1 – ranky pre superbloky
- R_2 – ranky pre bloky
- R_3 – tabuľka rankov pre krátke bitstringy

$$\text{rank}_1(S, i) = R_1[i/t_1] + R_2[i/t_2] + R_3[S[i' \ldots i' + t_2], i \bmod t_2]$$

- kde $i' \leftarrow i - i \bmod t_2$ (začiatok bloku)

$$\text{rank}_0(S, i) = i - \text{rank}_1(S, i)$$

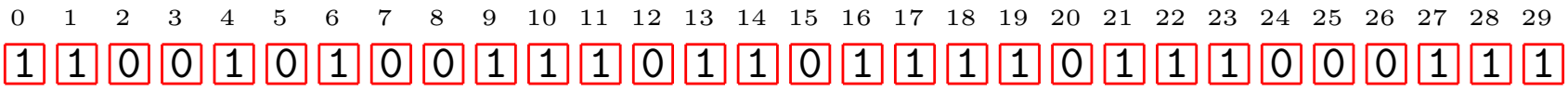
Rank

praktické riešenie:

- bloky dĺžky $7 \times 64 = 448$ bitov,
- ktoré poprekladáme predpočítanými rankami (jeden 64-bitový int)
- $\Rightarrow$ rank+blok je jedna cache line
- na rátanie použijeme popcnt (alebo vektorové inštrukcie?)
- iba 14% pamäte navyše

Select

Select

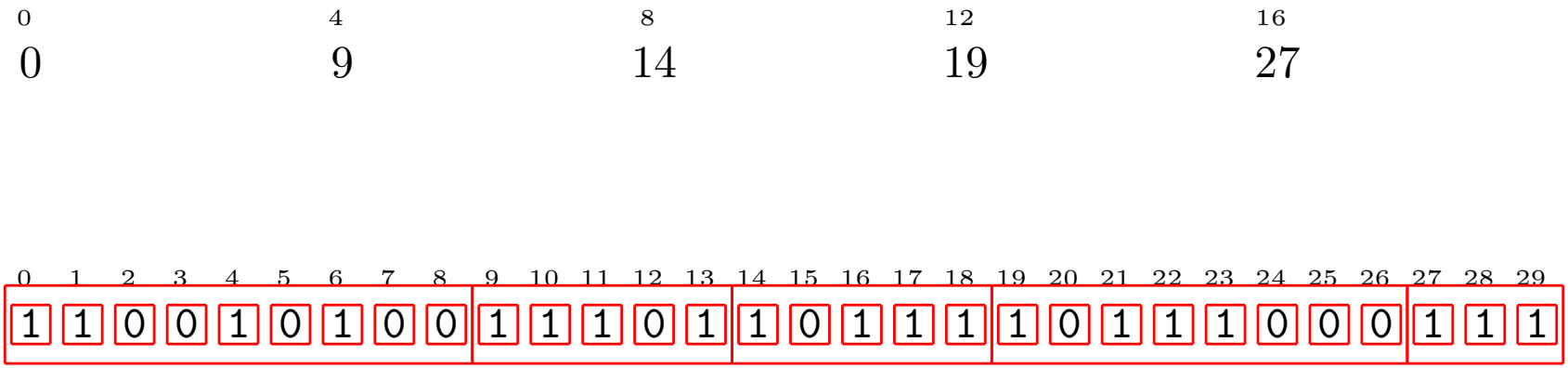

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29
1 1 0 0 1 0 1 0 0 1 1 1 0 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 1

Select

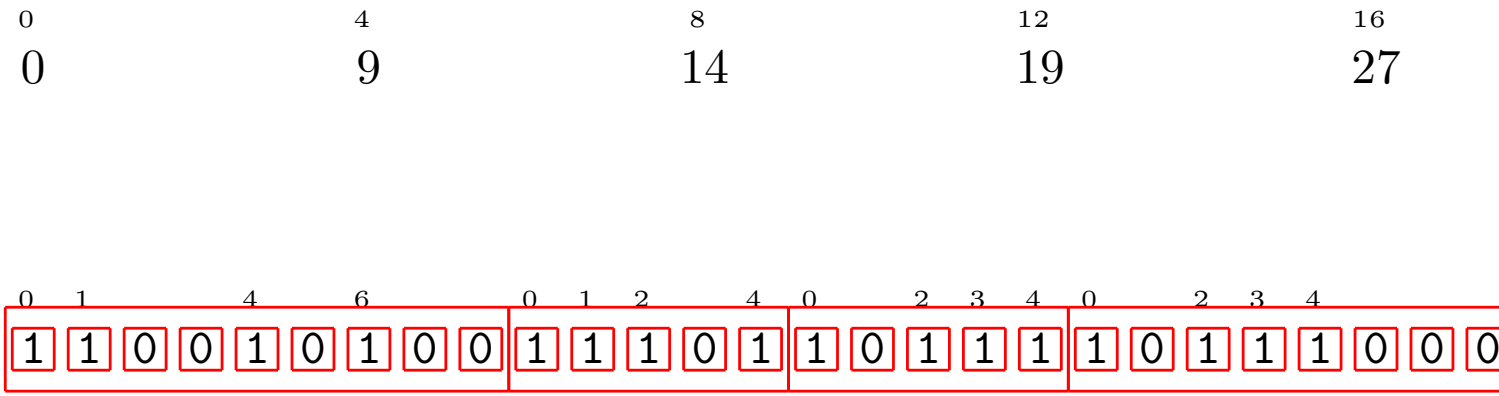
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
0	1	4	6	9	10	11	13	14	16	17	18	19	21	22	23	27	28	29

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
1	1	0	0	1	0	1	0	0	1	1	1	0	1	1	0	1	1	1	1	0	1	1	1	0	0	0	1	1	1

Select


0 4 8 12 16
0 9 14 19 27
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29
1 1 0 0 1 0 1 0 0 1 1 1 0 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 1

Select


0 4 8 12 16
0 9 14 19 27
0 1 4 6 0 1 2 4 0 2 3 4 0 2 3 4 0 1 2
1 1 0 0 1 0 1 0 0 1 1 1 0 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 1

Select

DŠ:

- predpočítame násobky $t_1 = \lg n \lg\lg n$ (superbloky) – $O(n/t_1 \cdot \lg n) = O(n/\lg\lg n) = o(n)$
- v rámci superbloku...

Select

DŠ:

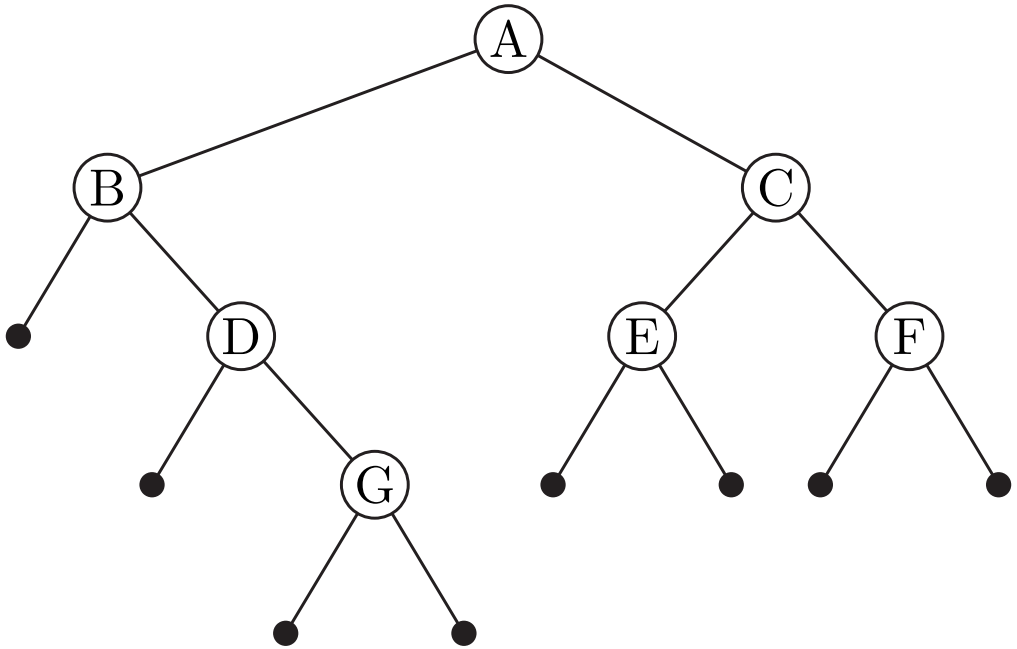
- predpočítame násobky $t_1 = \lg n \lg\lg n$ (superbloky) – $O(n/\lg\lg n) = o(n)$
  - pre dlhé (riedke) superbloky (dĺžky $\geq t_1^2$) predpočítame všetky pozície – $O(n/t_1^2 \cdot t_1 \cdot \lg n) = o(n)$
  - ostáva vyriešiť tie krátke, kde dĺžka je $< t_1^2$, takže na zápis pozície stačí $O(\lg\lg n)$ bitov
- to isté pre $t_2 = (\lg\lg n)^2$
- blokov veľkosti $< t_2^2$ je málo $\Longrightarrow$ tabuľka

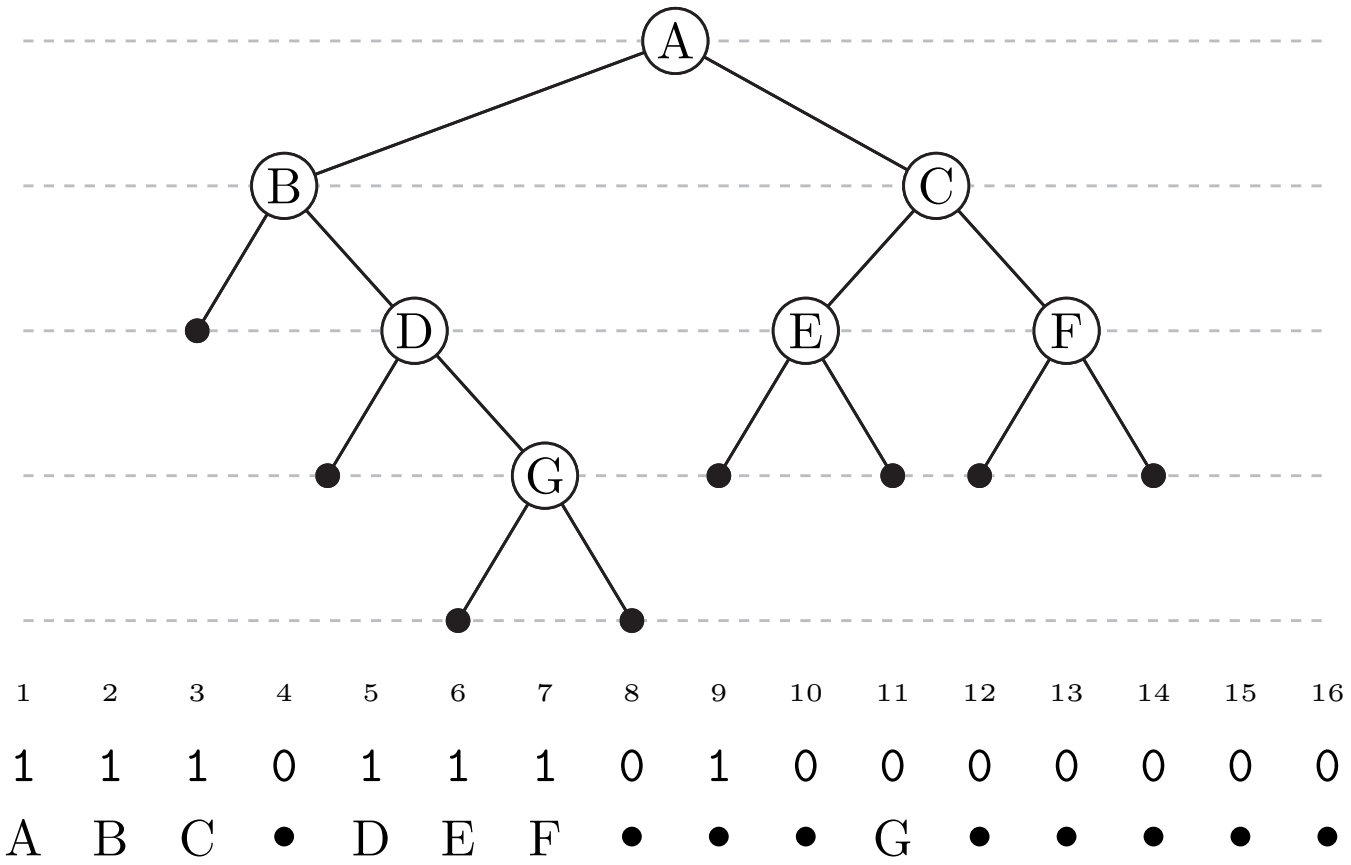
Select

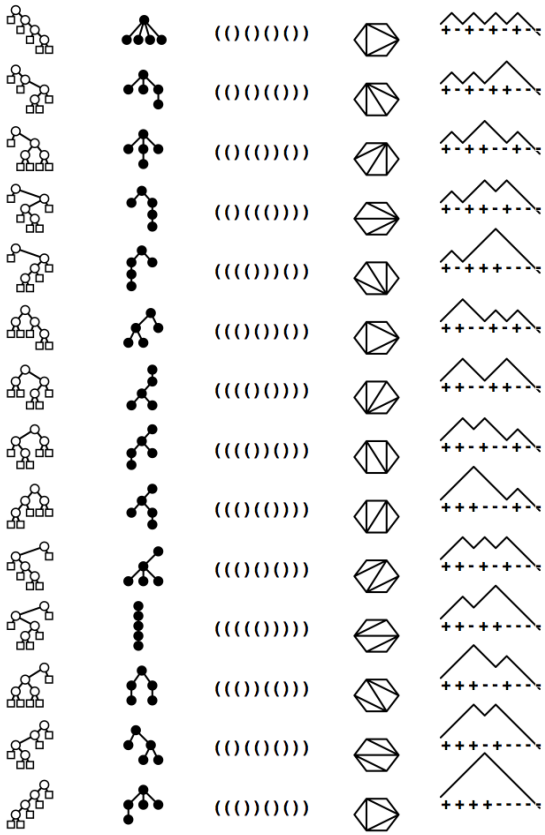
DŠ:

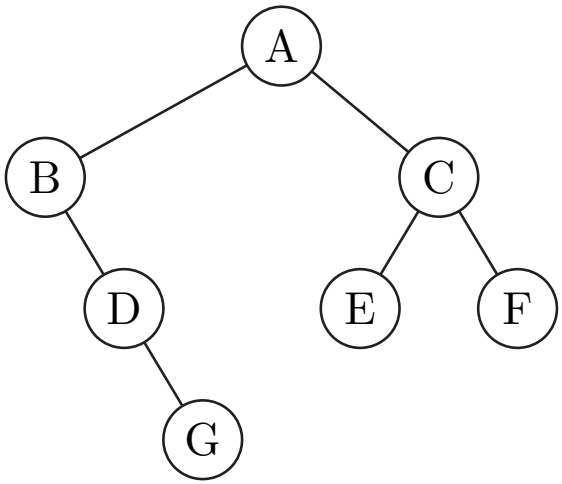
- predpočítame násobky $t_1 = \lg n \lg\lg n$ (superbloky) – $O(n/\lg\lg n) = o(n)$
  - pre dlhé (riedke) superbloky (dĺžky $\geq t_1^2$) predpočítame všetky pozície – $O(n/t_1^2 \cdot t_1 \cdot \lg n) = o(n)$
  - ostáva vyriešiť tie krátke, kde dĺžka je $< t_1^2$, takže na zápis pozície stačí $O(\lg\lg n)$ bitov
- to isté pre $t_2 = (\lg\lg n)^2$
- blokov veľkosti $< t_2^2$ je málo $\Longrightarrow$ tabuľka

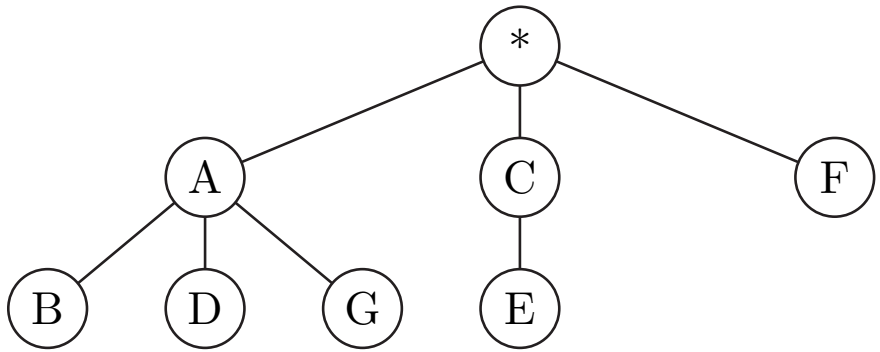
Úsporné stromy


A
B
C
D
E
F
G

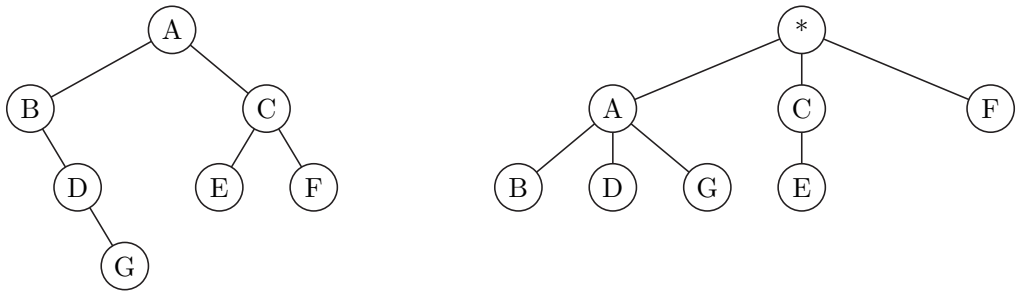

A
B
C
D
E
F
G
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
1 1 1 0 1 1 1 0 1 0 0 0 0 0 0 0
A B C • D E F • • • G • • • • •







(	(	(	)	(	)	(	)	)	(	(	)	)	(	)	)
*	A	B	B	D	D	G	G	A	C	E	E	C	F	F	*



(	(	(	)	(	)	(	)	)	(	(	)	)	(	)	)
*	A	B	B	D	D	G	G	A	C	E	E	C	F	F	*

- ľavý syn $\longleftrightarrow$ prvý syn $\longleftrightarrow$ ďalší znak (ak je (
- pravý syn $\longleftrightarrow$ ďalší sused $\longleftrightarrow$ znak za prísl.)
- rodič $\longleftrightarrow$ predch. sused alebo rodič $\longleftrightarrow$ ak je predch.), tak prísl. (, inak predch.

Väčšia abeceda?

DNA

- A, C, G, T
- rozdelíme na bloky, pre každý spočítame počet počet A/C/G/T po začiatok bloku

Wavelet strom

$$\Sigma = \{\text{␣}, \texttt{.}, \texttt{A}, \texttt{E}, \texttt{M}, \texttt{U}\}$$

MAMA␣MA␣EMU.␣EMA␣MA␣MAMU.

Wavelet strom

$$\Sigma_0 = \{\text{␣}, \texttt{.}, \texttt{A}\} \qquad \Sigma_1 = \{\texttt{E}, \texttt{M}, \texttt{U}\}$$

MAMA␣MA␣EMU.␣EMA␣MA␣MAMU.

Obr.: Rozdelíme abecedu na dve časti: $\Sigma = \Sigma_0 \cup \Sigma_1$.

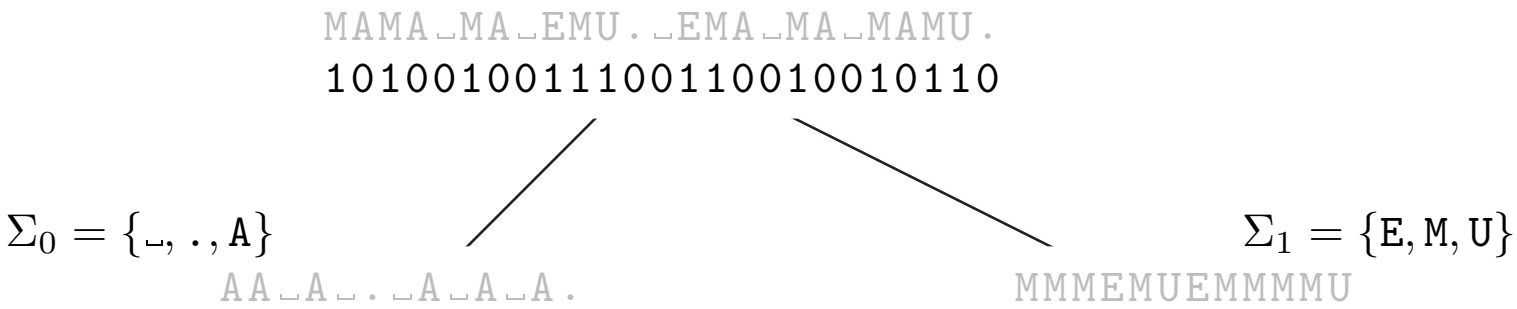
Wavelet strom

$$\Sigma_0 = \{\text{␣}, \texttt{.}, \texttt{A}\} \qquad \Sigma_1 = \{\texttt{E}, \texttt{M}, \texttt{U}\}$$

MAMA␣MA␣EMU.␣EMA␣MA␣MAMU.
10100100111001100100101 10

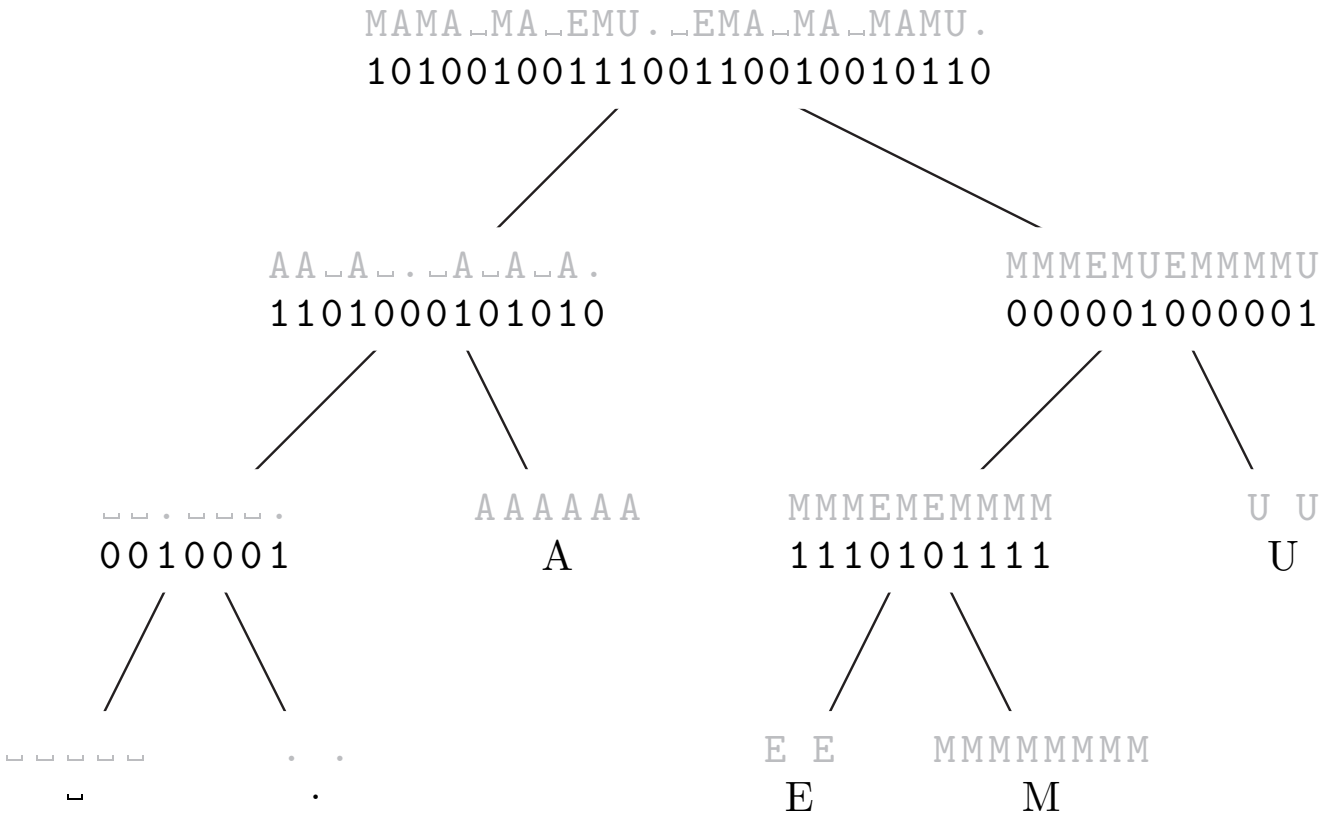
Obr.: Označíme, ktoré písmeno patrí kam: $B_i = j \iff S_i \in \Sigma_j$

Wavelet strom

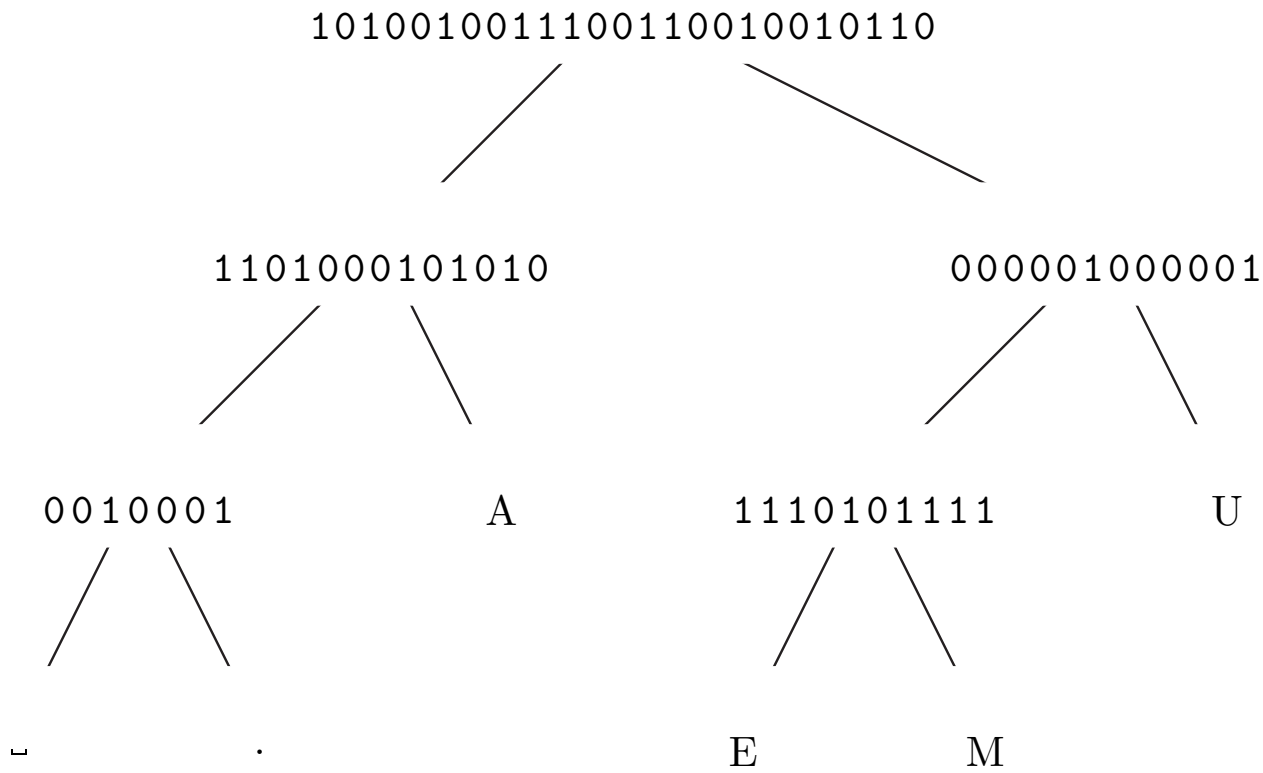


Obr.: $S_0 = S\backslash_{\Sigma_0}$ a $S_1 = S\backslash_{\Sigma_1}$ a rekurzívne pokračujeme.

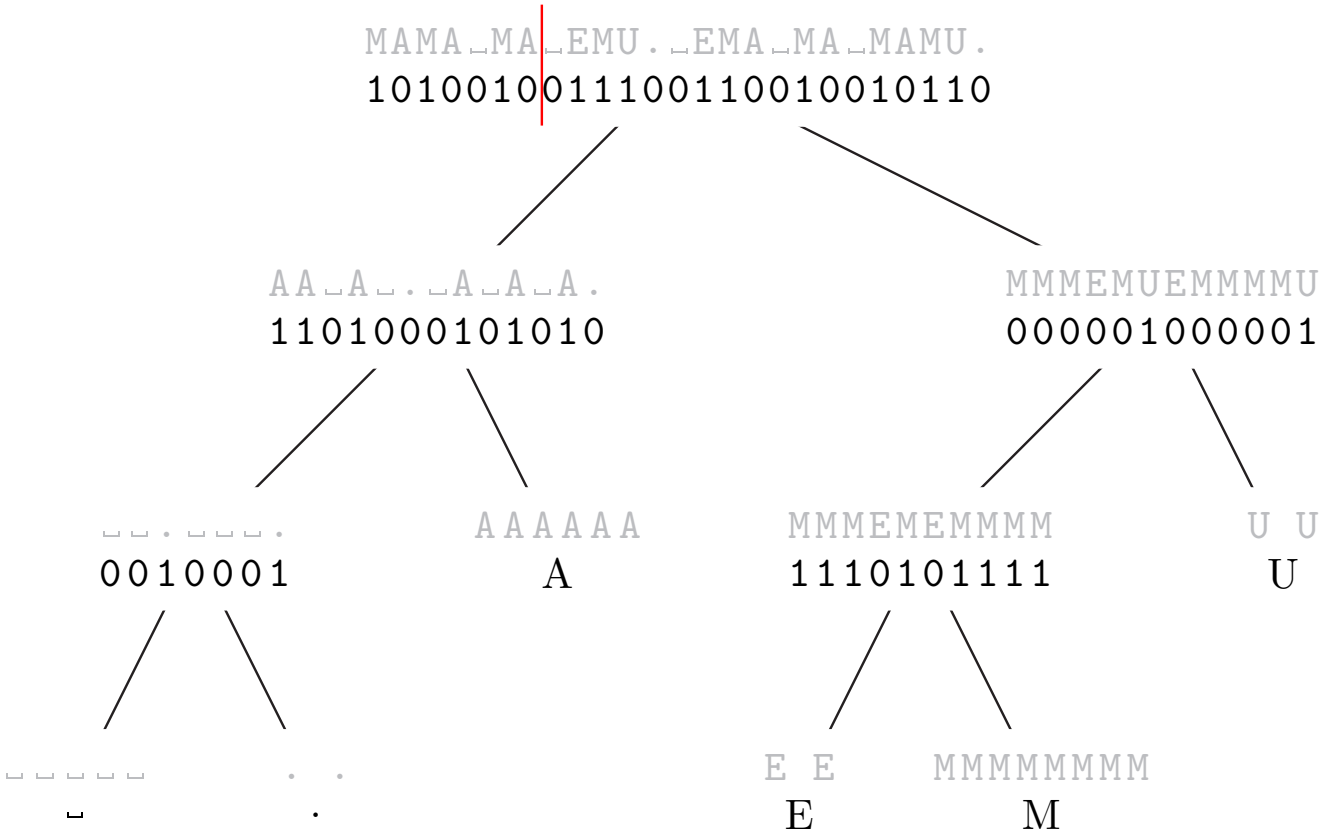
Wavelet strom



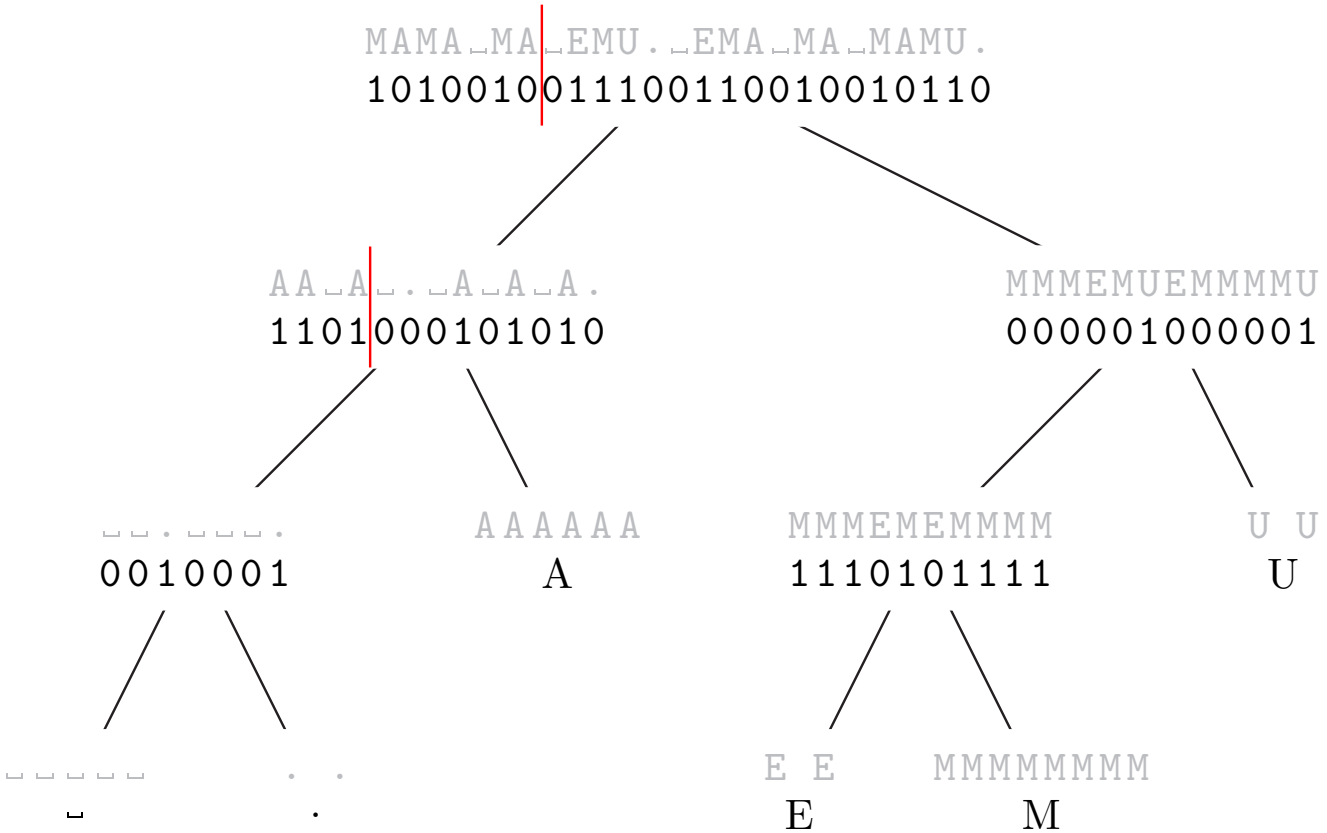
Wavelet strom



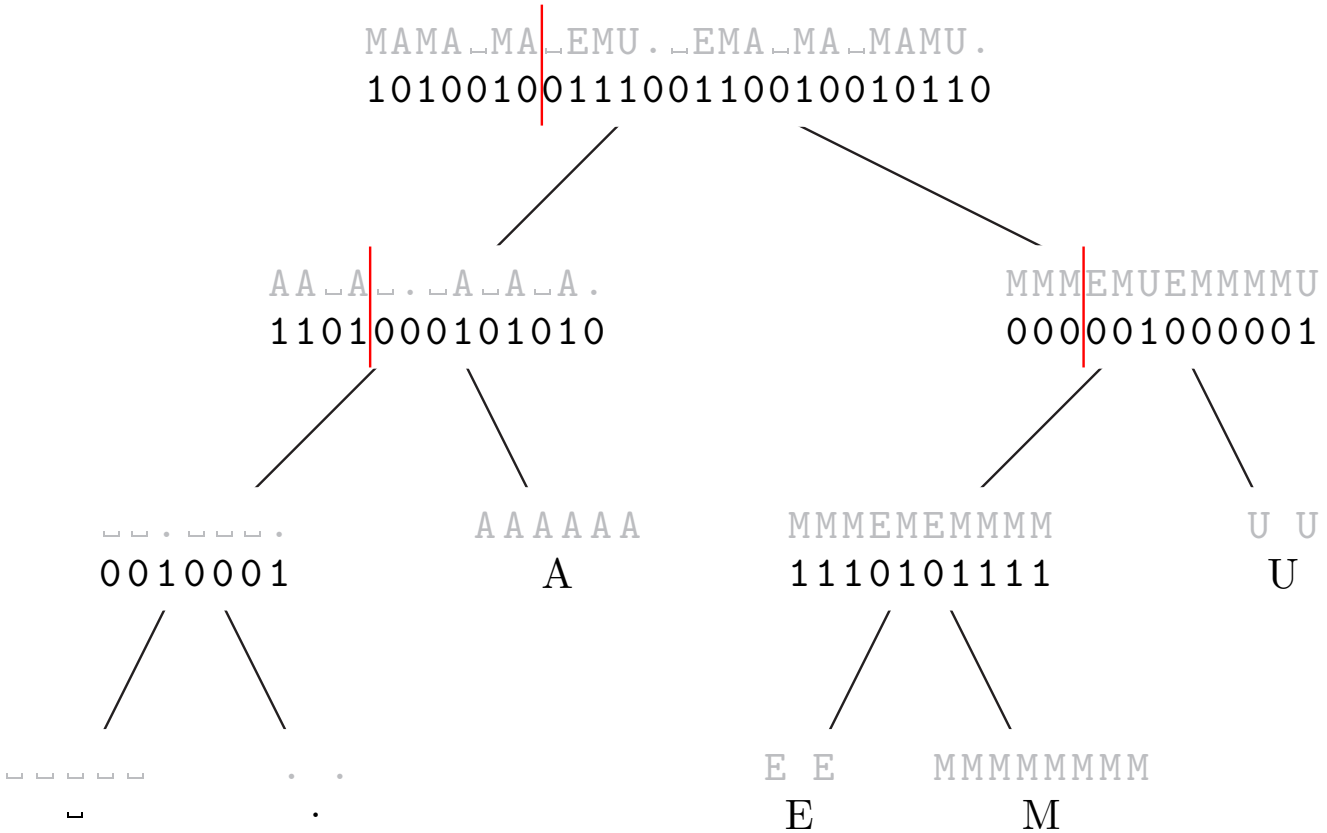
Wavelet strom



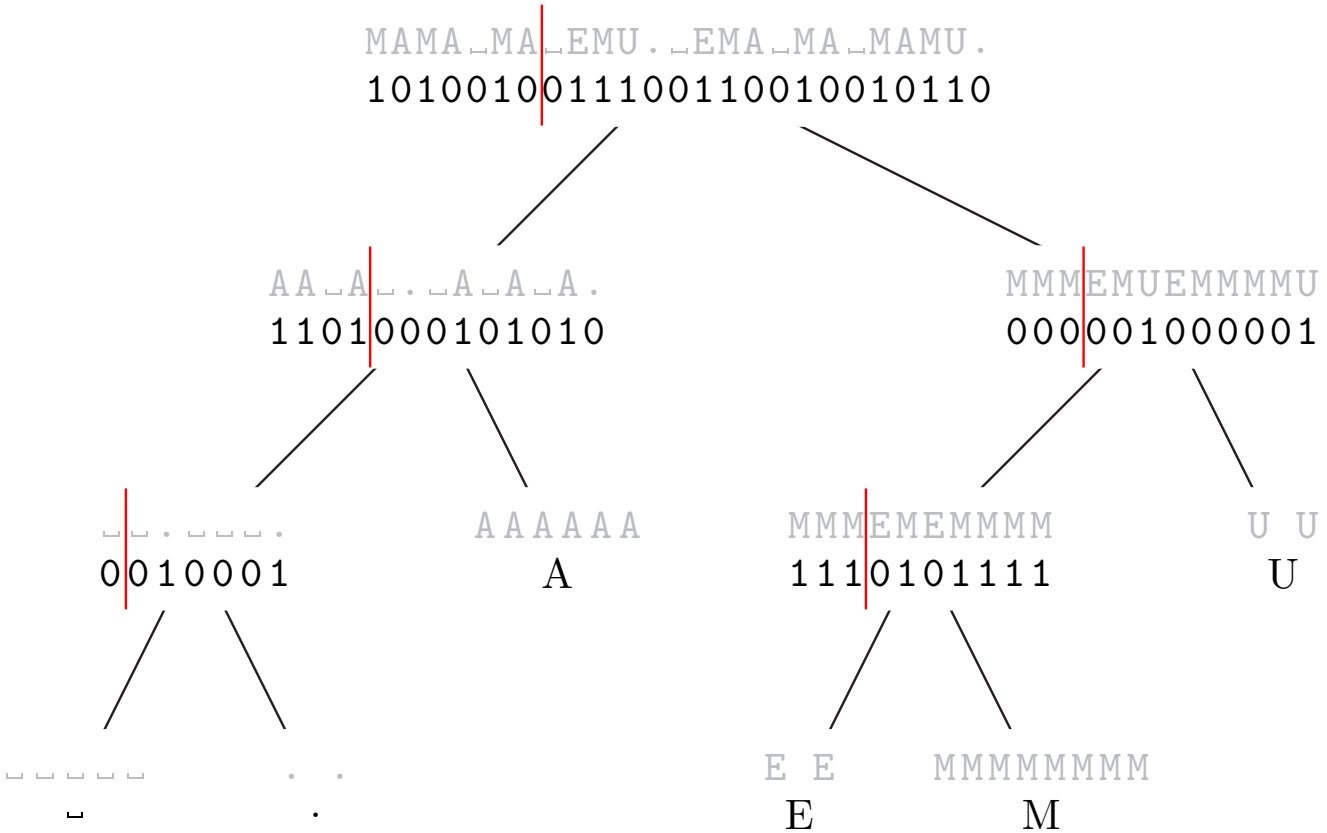
Wavelet strom



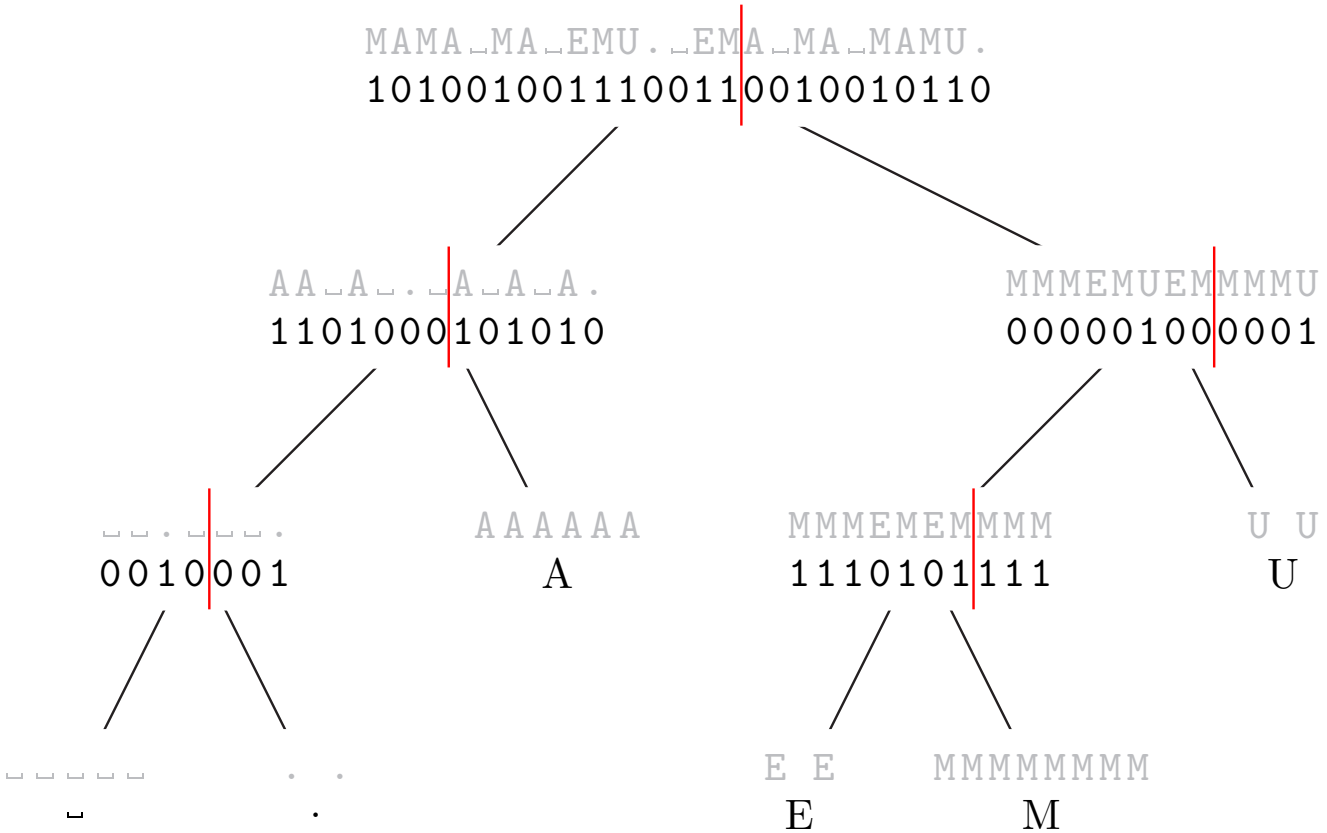
Wavelet strom



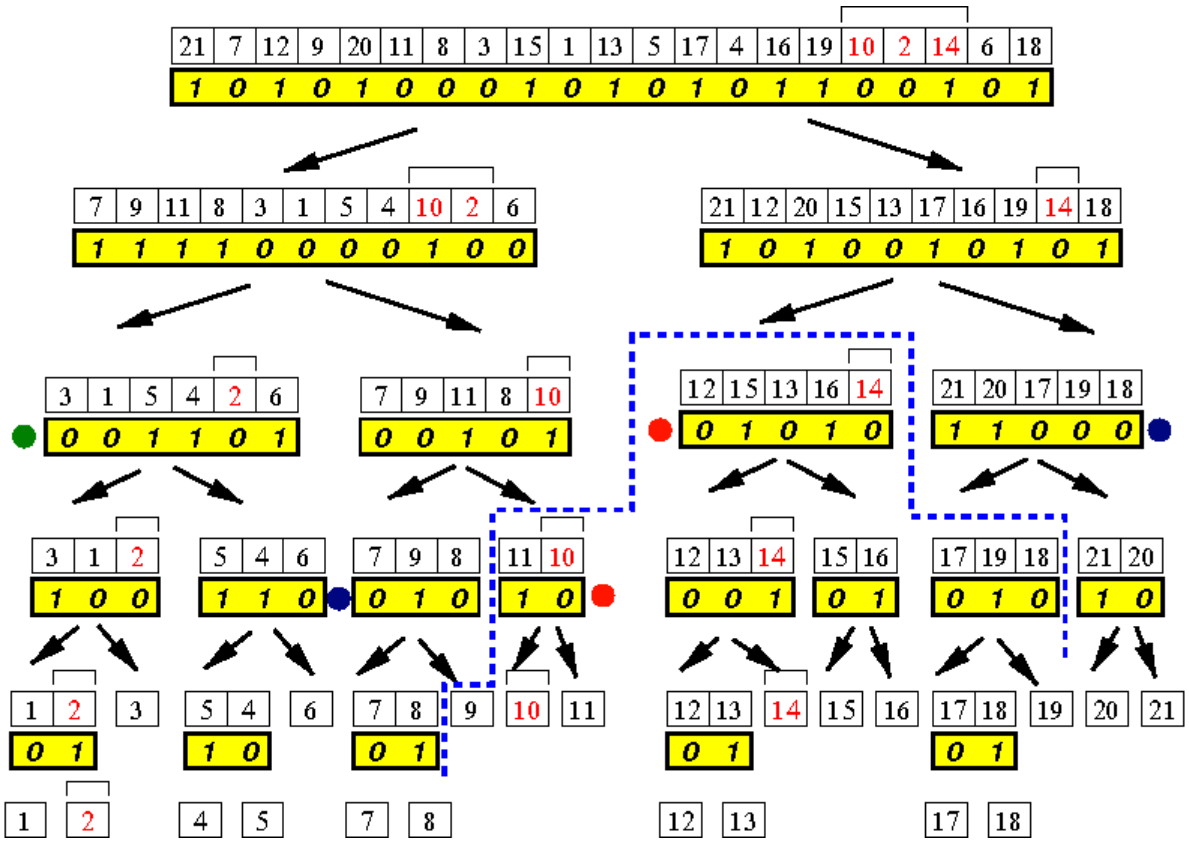
Wavelet strom



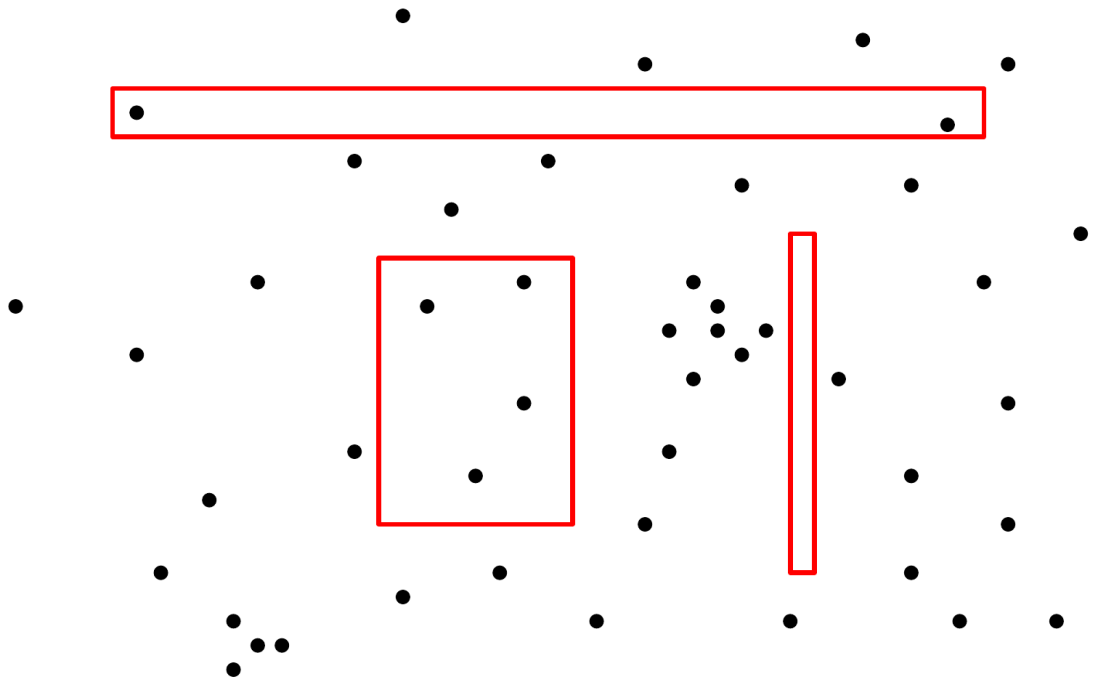
Wavelet strom



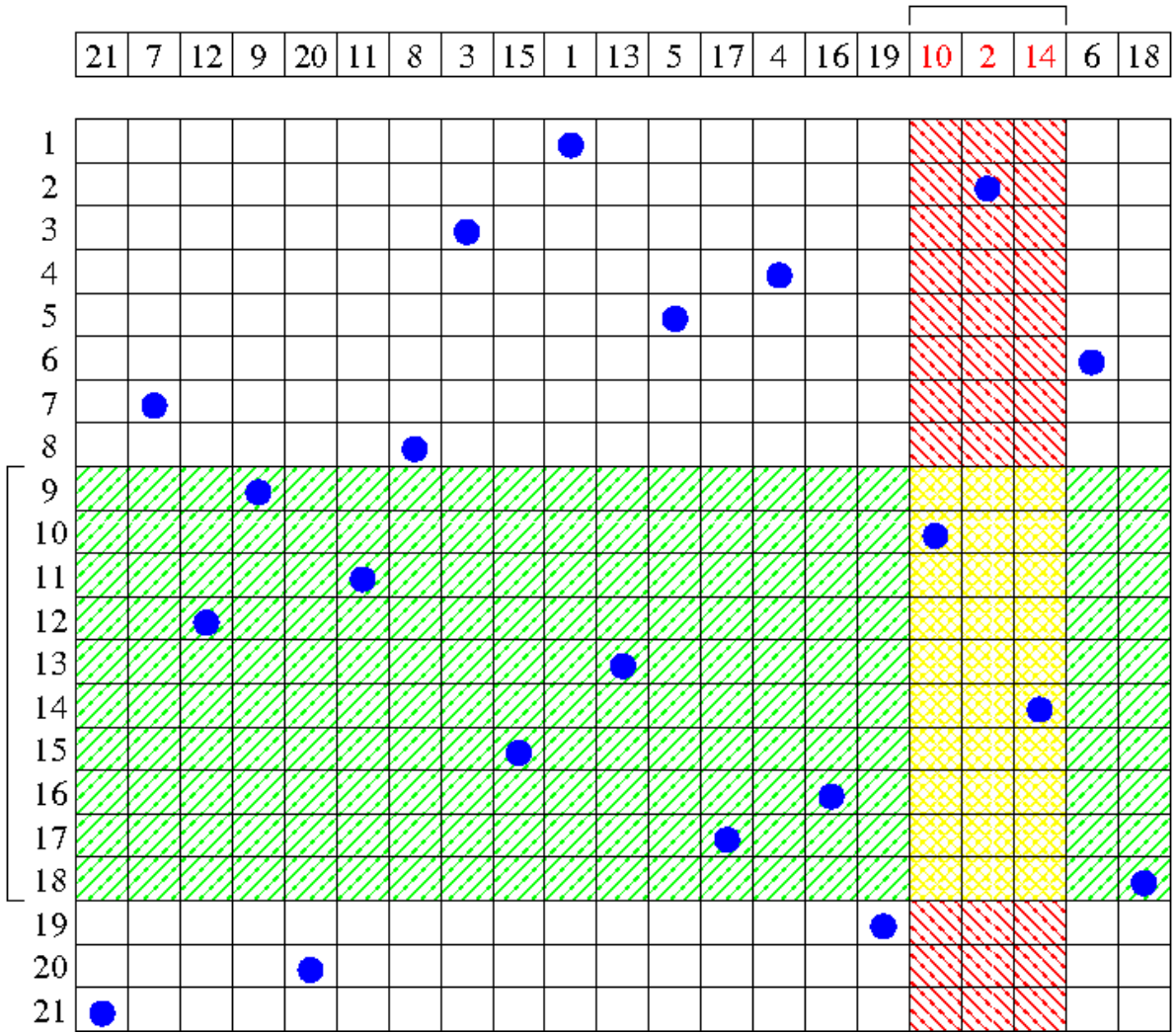
Wavelet strom



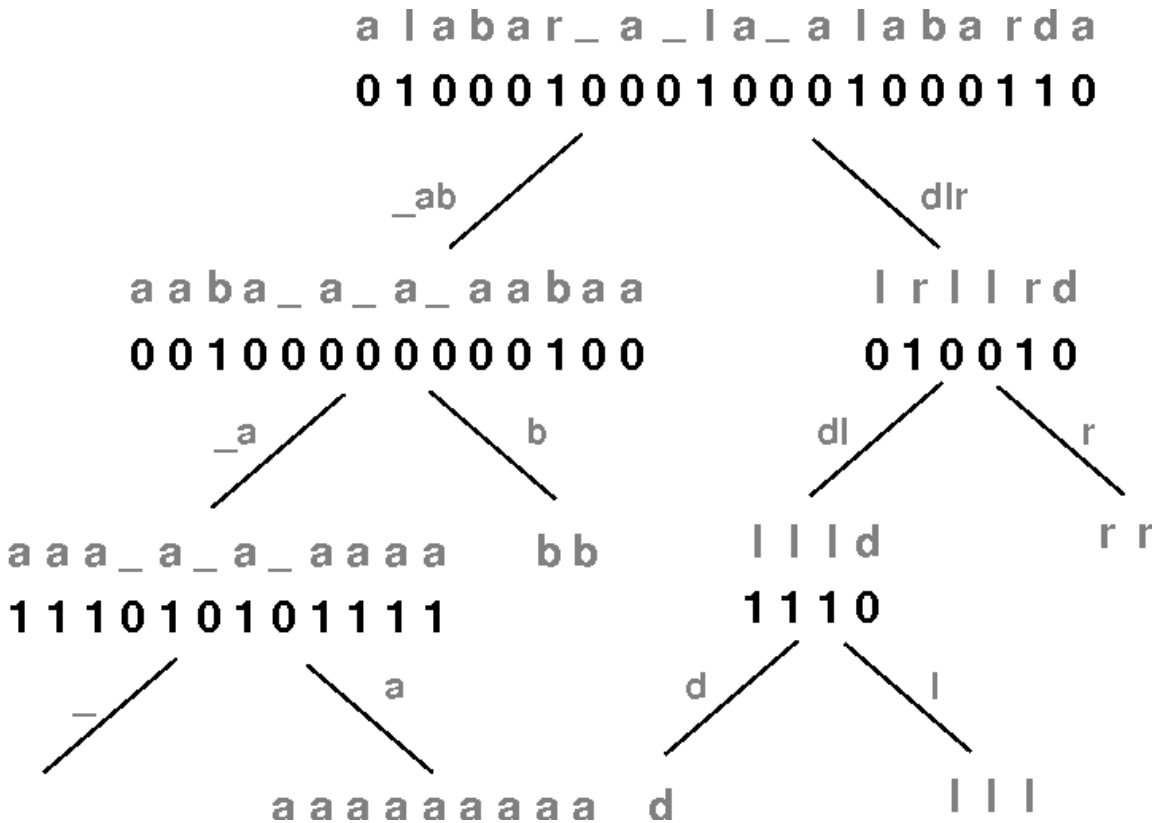
Wavelet strom



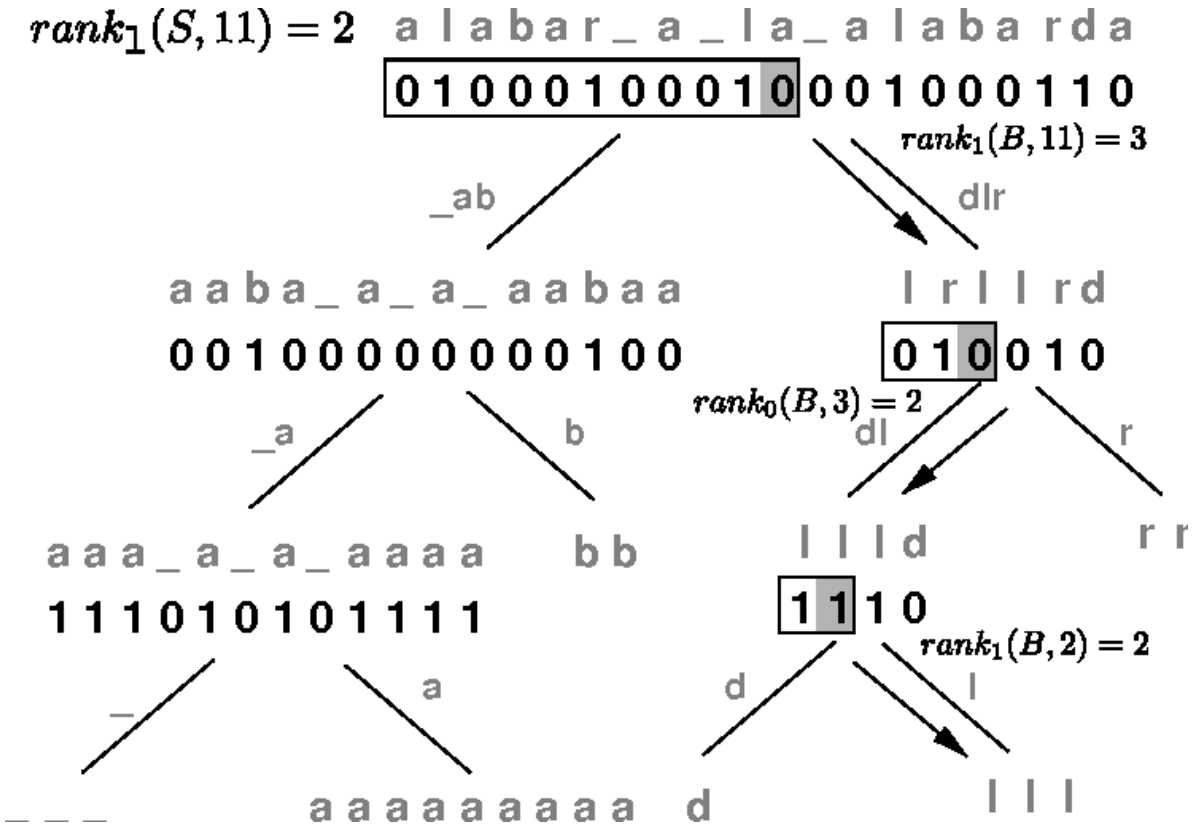
Wavelet strom



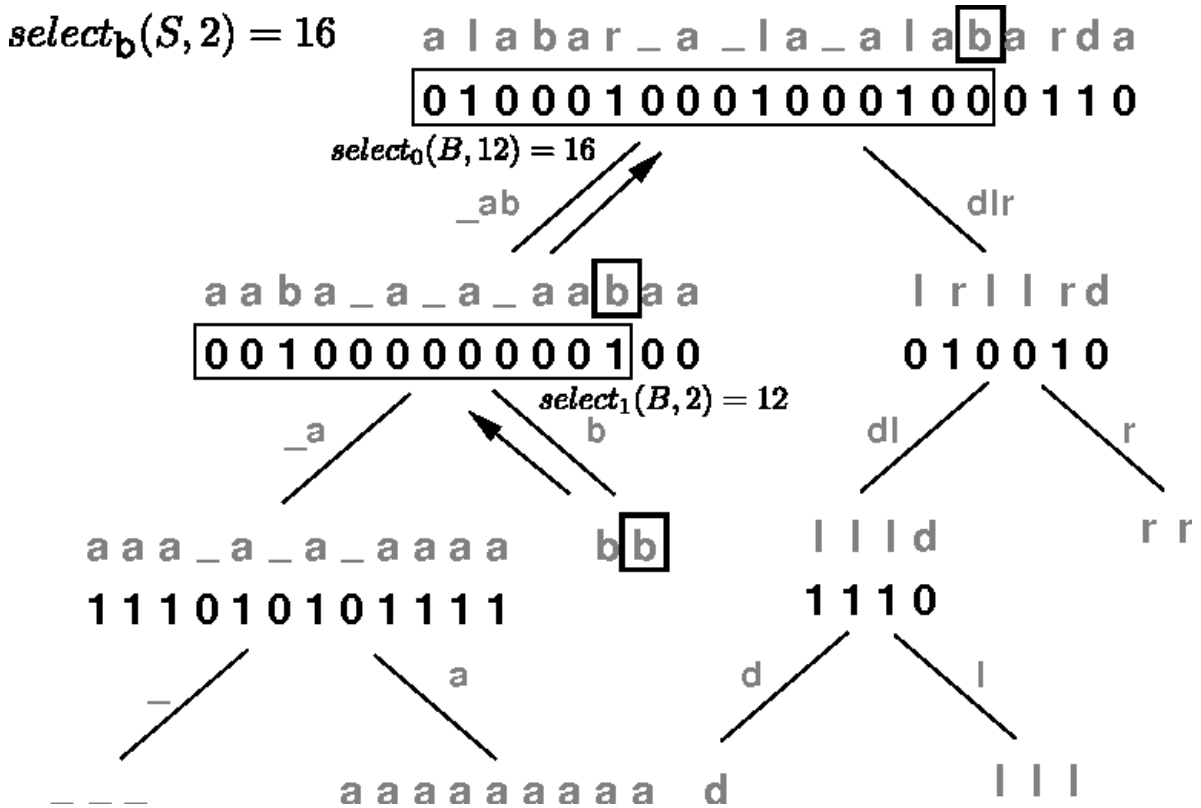
Wavelet strom



Wavelet strom



Wavelet strom



Wavelet strom