Organizačné poznámky

- Posledná šanca rozhodnúť sa pre projekt
- Do 15.4. si vyberte článok na prezentáciu
  – výber zapíšte v Moodli
  – vyberte zo zoznamu na stránke, alebo nejaký iný
  – prezentácie budú posledné dva týždne semestra
  – k článku sa vás môžem spýtať aj na skúške

Succint structure for rank and select

Bit vector A[0..n-1]

rank(i) = number of bits set to 1 in A[0..i]

select(i) = position of the i-th bit set to 1

Example:

i	0	1	2	3	4	5	6	7
$A[i]$	0	1	1	0	1	0	0	1

rank(3) = 2, rank(4) = 3

select(1) = 1, select(3) = 4

Goal:

rank, select in $O(1)$ time

structure needs $n + o(n)$ memory

we will concentrate on rank

Wavelet tree

$\Sigma_0 = \{\$, ., a\}$

$\Sigma_1 = \{e, m, u\}$

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
S[i]	e	m	a	.	m	a	.	m	a	m	u	.	m	a	m	a	.	m	a	.	e	m	u
B[i]	1	1	0	0	1	0	0	1	0	1	1	0	1	0	1	0	0	1	0	0	1	1	1
S0	a.a.a.aa.a.$																						
S1	emmmmummmemu																						

Suffix array

Array of suffixes in lexicographic order

(assume \$ $< a \quad \forall a \in \Sigma$)

i	0	1	2	3	4	5	6
$T[i]$	b	a	n	a	n	a	\$

i	$SA[i]$	Suffix
0	6	\$
1	5	a\$
2	3	ana\$
3	1	anana\$
4	0	banana\$
5	4	na\$
6	2	nana\$

Can be computed in $O(n)$

Burrows-Wheeler transform (BWT) 1994

T =banana$

Sort all rotations of the word lexicographically:

b	a	n	a	n	a	$
a	n	a	n	a	$	b
n	a	n	a	$	b	a
a	n	a	$	b	a	n
n	a	$	b	a	n	a
a	$	b	a	n	a	n
$	b	a	n	a	n	a

$	b	a	n	a	n	**a**
a	$	b	a	n	a	**n**
a	n	a	$	b	a	**n**
a	n	a	n	a	$	**b**
b	a	n	a	n	a	**$**
n	a	$	b	a	n	**a**
n	a	n	a	$	b	**a**

BWT = annb$aa

Can be computed using suffix array: BWT[i] = T[SA[i]-1] (or T[n] if SA[i]=0)

FM index

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
T[i]	e	m	a	.	m	a	.	m	a	m	u	.	m	a	m	a	.	m	a	.	e	m	u	$
SA[i]	23	19	16	3	11	6	18	15	2	5	13	8	0	20	17	14	1	4	12	7	21	9	22	10

T’[i]	u	a	a	a	u	a	m	m	m	m	m	m	$	.	.	a	e	.	.	.	e	a	m	m
r$(i)	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
r.(i)	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2	2	3	4	5	5	5	5	5
ra(i)	0	1	2	3	3	4	4	4	4	4	4	4	4	4	4	5	5	5	5	5	5	6	6	6
re(i)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	2	2	2	2
rm(i)	0	0	0	0	0	0	1	2	3	4	5	6	6	6	6	6	6	6	6	6	6	6	7	8
ru(i)	1	1	1	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2

x	$	.	a	e	m	u
C(x)	0	1	6	12	14	22

```
 0 23 $
 1 19 .emu$
 2 16 .ma.emu$
 3  3 .ma.mamu.mama.ma.emu$
 4 11 .mama.ma.emu$
 5  6 .mamu.mama.ma.emu$
 6 18 a.emu$
 7 15 a.ma.emu$
 8  2 a.ma.mamu.mama.ma.emu$
 9  5 a.mamu.mama.ma.emu$
10 13 ama.ma.emu$
11  8 amu.mama.ma.emu
12  0 ema.ma.mamu.mama.ma.emu$
13 20 emu$
14 17 ma.emu$
15 14 ma.ma.emu$
16  1 ma.ma.mamu.mama.ma.emu$
17  4 ma.mamu.mama.ma.emu$
18 12 mama.ma.emu$
19  7 mamu.mama.ma.emu$
20 21 mu$
21  9 mu.mama.ma.emu$
22 22 u$
23 10 u.mama.ma.emu$
```

Counting occurrences of P using FM index

```
1  L = 0; R = n;
2  for(i = m—1; i >= 0; i——) {
3    a = P[i];
4    L = C[a] + rank(a, L—1);
5    R = C[a] + rank(a, R) — 1;
6    if(L > R) return 0; // no occurrences
7  }
8  return R — L + 1;
```