Advanced Data Structures
Simon Gog – gog@kit.edu
Problem
Given a sequence $S[0, n - 1]$ of length $n$ over alphabet $\Sigma$ of size $\sigma$. Devise a data structure which efficiently supports the following operations:
- $\text{access}(i, S)$: Return $S[i]$ the element at position $i$ in $S$
- $\text{rank}(i, c, S)$: Return number of occurrences of element $c$ in $S[0, i - 1]$
- $\text{select}(i, c, S)$: Return the position of the $i$th occurrence of $c$ in $S$ (remember: $i$ is 1-indexed)
We present several results in this lecture. The following tables provides an overview. Note that we have omitted $O(\cdot)$ for time complexities.

<table>
<thead>
<tr>
<th></th>
<th>access($i, S$)</th>
<th>rank($i, c, S$)</th>
<th>select($i, c, S$)</th>
<th>space</th>
</tr>
</thead>
<tbody>
<tr>
<td>G-1</td>
<td>$\sigma \log \log \sigma$</td>
<td>log log $\sigma$</td>
<td>1</td>
<td>$n \log \sigma$</td>
</tr>
<tr>
<td>G-2</td>
<td>1</td>
<td>log log $\sigma$</td>
<td>1</td>
<td>$n \log \sigma + o(n \log \sigma)$</td>
</tr>
</tbody>
</table>

Results by Golynski, Munro, Rao (SODA 2006). A more careful analysis of G-1 results in entropy compressed space complexity.
Solution Overview

- (1) Divide sequence into blocks of length $\sigma$
- (2) Solve rank and select on block level
- (3) Solve in-block rank and select
- Step (1) and (2) are used in all structures
Conceptionally introduce a bitvector for each symbol
Concatenated in row major order to bitvector $A$
Size of $A$: $n\sigma$

$eyymmmmmm-$$eaarrrrrra$

$0000000000100000000000$
$0000000001110000000000$
$000000000000011100000001$
$100000000000010000000000$
$000011110000000000000000$
$00000000000001111110$
$011100000000000000000000$
Conceptionally introduce a bitvector for each symbol
Concatenated in row major order to bitvector $A$

Size of $A$: $n\sigma$
Conceptionally introduce a bitvector for each symbol
Concatenated in row major order to bitvector $A$
Size of $A$: $n\sigma$

```
  e y y y m m m m -- -- $ e a a r r r r r r a
$ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
- 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0
a 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1
e 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
m 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
r 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0
y 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```
Rank/Select on General Sequences

We have already see how to answer rank and select on bitvectors in constant time using and index of $o(n)$ bits of space. With these results we can answer rank/select on $S$:

\[
\begin{align*}
\text{rank}(i, c, S) & = \text{rank}(c \cdot n + i, 1, A) - \text{rank}(c \cdot n, 1, A) \quad (1) \\
\text{select}(i, c, S) & = \text{select}(\text{rank}(c \cdot n, 1, A) + i, 1, A) \quad (2)
\end{align*}
\]

We here assume that the symbols can be mapped to $[0, \sigma - 1]$. I.e. we map $\$ to 0, _ to 1, ...
We have already see how to answer rank and select on bitvectors in constant time using and index of $o(n)$ bits of space. With these results we can answer rank/select on $S$:

\[
\begin{align*}
\text{rank}(i, c, S) &= \text{rank}(c \cdot n + i, 1, A) - \text{rank}(c \cdot n, 1, A) \quad (1) \\
\text{select}(i, c, S) &= \text{select}(\text{rank}(c \cdot n, 1, A) + i, 1, A) \quad (2)
\end{align*}
\]

We here assume that the symbols can be mapped to $[0, \sigma - 1]$. I.e. we map $\$ \rightarrow 0, \_ \rightarrow 1, \ldots$

Unfortunately, $A$ uses a lot of space.
We have already see how to answer rank and select on bitvectors in constant time using and index of $o(n)$ bits of space. With these results we can answer rank/select on $S$:

\[
\begin{align*}
\text{rank}(i, c, S) &= \text{rank}(c \cdot n + i, 1, A) - \text{rank}(c \cdot n, 1, A) \\
\text{select}(i, c, S) &= \text{select}(\text{rank}(c \cdot n, 1, A) + i, 1, A)
\end{align*}
\]

We here assume that the symbols can be mapped to $[0, \sigma - 1]$. I.e. we map $\$ to 0, \_ to 1,\ldots

Unfortunately, $A$ uses a lot of space. But we can compress it:
### Rank/Select on General Sequences

- Divide $A$ into blocks of length $\sigma$
- Count the number of ones in each block of $A$
- Store the counts in array $C$ of length $n$ (takes $n \log \sigma$ bits)

| e | y | y | y | y | m | m | m | m | m | m | m | m | m | – | – | – | $\$ | e | a | r | r | r | r | r | r | r | a |
| $\$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| – | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| a | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| e | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| m | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| r | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| y | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Simon Gog:**
Advanced Data Structures

**Institute of Theoretical Informatics**
Algorithmics
Divide $A$ into blocks of length $\sigma$

Count the number of ones in each block of $A$

Store the counts in array $C$ of length $n$ (takes $n \log \sigma$ bits)
Divide A into blocks of length $\sigma$

Count the number of ones in each block of A

Store the counts in array $C$ of length $n$ (takes $n \log \sigma$ bits)
Rank/Select on General Sequences

\[ C = 0 \ 1 \ 0 \ 0 \ 3 \ 0 \ 0 \ 1 \ 2 \ 1 \ 1 \ 0 \ 3 \ 1 \ 0 \ 0 \ 0 \ 5 \ 3 \ 0 \ 0 \]

- The sum of all entries in \( C \) is \( n \)
- We can concatenate all values unary encoded into bitvector \( B \). In our example: \( B = 101110001110100101011000101111000001000111 \)
- Size of \( B \) is \( 2n \) bits
- We can perform operations on blocks by adding one select structure
Rank/Select on General Sequences

\[ C = 0 \ 1 \ 0 \ 0 \ 3 \ 0 \ 0 \ 1 \ 2 \ 1 \ 1 \ 0 \ 3 \ 1 \ 0 \ 0 \ 0 \ 5 \ 3 \ 0 \ 0 \]

- The sum of all entries in \( C \) is \( n \)
- We can concatenate all values unary encoded into bitvector \( B \). In our example: \( B = 101110001110100101011000101111000001000111 \)
- Size of \( B \) is \( 2n \) bits
- We can perform operations on blocks by adding one select structure
Rank/Select on General Sequences

\[ C = 0 1 0 0 3 0 0 1 2 1 1 0 3 1 0 0 0 5 3 0 0 \]

- The sum of all entries in \( C \) is \( n \)
- We can concatenate all values unary encoded into bitvector \( B \). In our example: \( B = 1 0 1 1 1 0 0 0 1 1 1 0 1 0 0 1 0 1 0 1 1 0 0 0 1 0 1 1 1 1 0 0 0 0 0 1 0 0 0 1 1 1 \)
- Size of \( B \) is \( 2n \) bits
- We can perform operations on blocks by adding one select structure
The sum of all entries in $C$ is $n$

We can concatenate all values unary encoded into bitvector $B$. In our example: $B = 101110001110100101100101111000101000111$

Size of $B$ is $2n$ bits

We can perform operations on blocks by adding one select structure
The sum of all entries in $C$ is $n$

We can concatenate all values unary encoded into bitvector $B$. In our example: $B = 101110001110100101101100010111100001000111$

Size of $B$ is $2n$ bits

We can perform operations on blocks by adding one select structure
Rank/Select on General Sequences

\[
B = 101110001110100101011000101111000001000111
\]

- \( \text{rank}'(\sigma i, A) = \text{select}(i, 1, B) + 1 - i \)
- \( \text{select}'(i, A) = \text{rank}(\text{select}(i, 0, B), 1, B) = \text{select}(i, 0, B) + 1 - i \)
Rank/Select on General Sequences
In-block rank and select (G-1)

- For each block $A_j$, we store the positions in the range $[0, \sigma - 1]$ of the set bits in increasing order in an array $E_j$
- Total space: $n \log \sigma$

Solve select
Block $x = select'(i, 1, A)$ contains the $i$-th one. There are $y = rank'(\sigma x, 0, A)$ ones before block $x$ \Rightarrow
select(i, 1, S) = x \cdot \sigma + E_x[i - y]$

Solve rank
$i$ with $j = \left\lfloor \frac{i}{\sigma} \right\rfloor$ and $r = i - j \cdot \sigma$
rank($i, 1, S$) = rank'($i \cdot \sigma$, $A$) + $\max\{\{k | E_j[k] < r\} \cup \{-1\}\} + 1$
Use $y$-fast trie for second part to get $O(\log \log \sigma)$ time
Divide $S$ in chunks of size of size $\sigma$.

In each chunk $C$: For each $c \in \Sigma$ (in lex. order) write its occurrences in $C$. We get a permutation $\pi$.

Also store a bitvector $X$ which contains the number of occurrences decoded in unary.

\[ S = e y y y m m m m m m - - - - - $ e a a a r r r r r r a \]

\[ \pi = 0 4 5 6 1 2 3 4 1 2 3 6 5 0 0 6 1 2 3 4 5 \]

\[ X = 1 1 1 0 1 0 0 0 1 1 0 0 0 1 0 1 0 0 0 1 0 1 0 1 0 1 1 1 1 1 0 0 1 1 1 0 0 0 0 0 0 1 1 \]

\[ $-a e m r y $ - a e m r y$- a e m \]

\[ \text{ry} \]
Rank/Select on General Sequences

In-block access, rank, and select (G-2)

- **select**\((i, c, S)\): First we determine by rank and select on \(A\) chunk \(x\) and the argument \(j\) for select on \(C_x\)

- \(select(j, c, C_x) = \pi_X[select(c, 1, X) + j - c]\)

\[
S = e y y y m m m m m m - - - - S e a a r r r r r a
\]

\[
\pi = \begin{array}{cccccccc}
0 & 4 & 5 & 6 & 1 & 2 & 3 & 4 & 1 & 2 & 3 & 6 & 5 & 0 & 0 & 6 & 1 & 2 & 3 & 4 & 5
\end{array}
\]

\[
X = 1 1 1 0 1 0 0 0 1 1 0 0 0 1 0 1 0 0 0 1 0 1 0 1 0 1 1 1 1 1 0 0 1 1 1 0 0 0 0 0 1 1
\]

\[
S-a e m r y S - a e m r y S-a e m r y
\]
Rank/Select on General Sequences

In-block access, rank, and select (G-2)

- \( y = \pi^{-1}(i) \) tells us the corresponding 0 in \( X \)
- Ones before \( y \) in \( X \) the corresponding character
- I.e. \( \text{select}(y, 0, X) = y - 1 \)

\[
\[ S = \text{e y y y m m m m m - - - - - e a a a r r r r r a} \\
\pi = \begin{array}{cccccccc}
0 & 4 & 5 & 6 & 1 & 2 & 3 & \\
4 & 1 & 2 & 3 & 6 & 5 & 0 & \\
0 & 6 & 1 & 2 & 3 & 4 & 5 & \\
\end{array}
\]

\[
X = 11101000110001010001010101111110011100000111 \\
\$-a e m r y $ - a e m r y $- a e m r y $
Use $X$ to select the range $[sp, ep]$ of position of $c$ in $\pi$

Solve predecessor query on $\pi[sp..ep]$