Reviewing: Golynski, Munro, Rao: Rank/Select Operations on Large Alphabets: a Tool for Text Indexing

Simon Gog

May 21, 2012
**Time complexities of basic operations on sequences**

Given sequence $T$ of length $n$ over alphabet $\Sigma$ of length $\sigma$.

<table>
<thead>
<tr>
<th></th>
<th>$\text{access}(T, i)$</th>
<th>$\text{rank}(T, i, c)$</th>
<th>$\text{select}(T, i, c)$</th>
<th>space</th>
</tr>
</thead>
<tbody>
<tr>
<td>WT</td>
<td>$\log \sigma$</td>
<td>$\log \sigma$</td>
<td>$\log \sigma$</td>
<td>$n \log \sigma + o(n \log \sigma)$</td>
</tr>
<tr>
<td>G-1</td>
<td>$\sigma \cdot \log \log \sigma$</td>
<td>$\log \log \sigma$</td>
<td>$1$</td>
<td>$nH_0 + O(n)$*</td>
</tr>
<tr>
<td>G-2</td>
<td>$\log \log \sigma$</td>
<td>$\log \log \sigma$</td>
<td>$1$</td>
<td>$n \log \sigma + o(n \log \sigma)$**</td>
</tr>
<tr>
<td>G-2a</td>
<td>$1$</td>
<td>$\log \log \sigma \cdot \log \log \log \sigma$</td>
<td>$\log \log \sigma$</td>
<td>$n \log \sigma + o(n \log \sigma)$**</td>
</tr>
</tbody>
</table>

We have omitted $O(\cdot)$ at the specification of the time complexities.

* actually $2n + o(n)$

** $4n + o(n)$ hidden in $o(n \log \sigma)$
Solution overview

- (1) Divide sequence into blocks of length $\sigma$.
- (2) Calculate rank and select on the block level.
- (3) Calculate in-block rank and select.
- Step (1) and (2) are used in all solutions
Relation between binary rank/select and general rank/select

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

```
  e y y y m 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 1 0 0 0 0 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 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 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 0 0 0
r 0 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 0 0 0 0
```
Relation between binary rank/select and general rank/select

If we can answer rank/select on $A$ in constant time, we can answer it on $T$ as well.

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

But: $A$ uses too much space!
Compressing $A$

- Divide $A$ into blocks of length $\sigma$
- Count the number of ones in each block $A$
- Resulting array is $C$ of length $n$, so $|C| = n \log \sigma$ bits.

| e | y | y | y | m | m | m | m | - | - | - | $\$ | e | a | a | r | r | r | r | r | a |
| $\$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| - | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| a | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 2 | 0 | 0 | 1 |
| e | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| m | 0 | 0 | 0 | 3 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 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 | 1 | 1 | 5 | 1 | 1 | 0 |
| y | 0 | 1 | 1 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
Compressing $A$

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

- The sum of all entries in $C$ is $n$.
- So store it with unary code in array $B$ with \( \Rightarrow |B| = 2n \) bits

\[ B = 101110001110100101011000101111000001000111 \]

- Now $C$ is compressed. But how do we answer rank and select with $B$?
- By adding a select data structure to $B$ we can navigate to blocks in $A$!
- We jump to block $i$ in $A$ by doing $\text{select}(B, i, 1)$
- \( \text{rank}'(A, \sigma i) = \text{rank}(B, \text{select}(B, i, 1), 0) = \text{select}(B, i, 1) + 1 - i \)
Rank and select on blocks $A$

$$B = 101110001111010010101011000101111000001000111$$

1. $\text{rank}'(A, \sigma i) = \text{select}(B, i, 1) + 1 - i$
2. $\text{select}'(A, i) = \text{rank}(B, \text{select}(B, i, 0), 1) = \text{select}(B, i, 0) + 1 - i$
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: $n \log \sigma$.

Answering select

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

Answering rank

- $i$ with $j = \lfloor \frac{i}{\sigma} \rfloor$ and $r = i - j \cdot \sigma$  
  $rank(A, i, 1) = rank'(A, i \cdot \text{sigma}) + \max\{\{k \mid E_j[k] < r\} \cup \{-1\}\} + 1$

  Use $y$-fast trie for second part to get $O(\log \log \sigma)$ time
Solution for rank/select and access (G-2)

- Divide T 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.

$$\pi = \begin{array}{cccccccccccccccccccccccc}
0 & 4 & 5 & 6 & 1 & 2 & 3 & 4 & 1 & 2 & 3 & 6 & 5 & 0 & 0 & 6 & 1 & 2 & 3 & 4 & 5
\end{array}$$

$$X = \begin{array}{cccccccccccccccccccccccc}
0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1
\end{array}$$
Solution for rank/select and access (G-2)

- $\text{select}(T, i, c)$: First we determine by $\text{rank}$ and $\text{select}$ on $A$ chunk $x$ and the argument $j$ for $\text{select}$ on $C_x$.
- $\text{select}(C_x, j, c) = \pi_X[\text{select}(X, c, 1) + j - c]

$$
\begin{array}{cccccccccccc}
\pi = & 0 & 4 & 5 & 6 & 1 & 2 & 3 & 4 & 1 & 2 & 3 & 6 & 5 & 0 & 0 & 6 & 1 & 2 & 3 & 4 & 5 \\
X = & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
\end{array}
$$
Solution for rank/select and access (G-2)

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

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

\[
X = 111101000110001010001010101111100111000001 \\
$_{a\ e\ m\ r\ y}$ $_{a\ e\ m\ r\ y}$ $._{a\ e\ m\ r\ y}$ $._{a\ e\ m\ r}$
\]
Solution for rank/select and access (G-2)

- Use $X$ to select the range $[sp..ep]$ of position of $c$ in $\pi$.
- Solve predecessor query on $\pi[sp..ep]$

$$\pi = \begin{array}{cccccccccccccc}
0 & 4 & 5 & 6 & 1 & 2 & 3 & 4 & 1 & 2 & 3 & 6 & 5 & 0 & 0 & 6 & 1 & 2 & 3 & 4 & 5
\end{array}$$

$$X = \begin{array}{cccccccccccccccccccc}
1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{array}$$