Please sign guestbook (www.langmead-lab.org/teaching-materials) to tell me briefly how you are using the slides. For original Keynote files, email me (ben.langmead@gmail.com).
Clark's select

Unlike rank:

Chunks are defined by # 1s, not # bits

Two layers of special-casing on sparsity

Answer is an offset into the bitvector — not a rank — so tables will hold offsets
Bitvectors

\[ B \]

\[
\begin{array}{c}
0 \\
1 \\
1 \\
0 \\
1 \\
0 \\
0 \\
1 \\
0 \\
1
\end{array}
\]

- \( B \cdot \text{select}_0(3) = 7 \)
- \( B \cdot \text{select}_1(5) = 12 \)
- \( B \cdot \text{rank}_0(5) = 2 \)
- \( \mathbb{B} \cdot \text{select}_1 \)
- \( \mathbb{B} \cdot \text{rank}_1 \)
- \( [0, n) \)
- \( [0, m) \)
Clark's select

Split into $\log^2 n$-weight chunks

Sparse ($\geq \log^4 n$-length)

Lookup table for each 1-bit

Dense ($< \log^4 n$-length)

Split into $\sqrt{\log n}$-weight sub-chunks

Sparse ($\geq 1/2 \log n$-length)

Lookup table for each 1-bit

Dense ($< 1/2 \log n$-length)

Lookup table for all possible sub-chunks
Clark's select

\[ T : \]

Split the string into chunks each containing \( \log^2 n \) 1-bits

Larger chunks are \textit{sparse}; 1’s spread out

Shorter chunks are \textit{dense}; 1’s packed together
Clark's select

Each chunk contains $\log^2 n$ 1-bits

We store offset of each chunk start

This takes:

$$O \left( \frac{n}{\log^2 n \log n} \right) = O \left( \frac{n}{\log n} \right) = \tilde{o}(n) \text{ bits}$$
Clark's select

Chunks $\geq \log^4 n$ bits in length are \textbf{sparse}, others \textbf{dense}

$\log^4 n$ is square of the \# of set bits per chunk, $\log^2 n$
Clark's select: sparse case

Store answers to $B \cdot \text{select}_1$ for 1-bits in all \textbf{sparse} chunks

\[
O \left( \frac{n}{\log^4 n} \cdot \log n \cdot \log^2 n \right)
\]

Max \# sparse chunks \# bits to store 1 answer \# answers per chunk

\[
= O \left( \frac{n}{\log n} \right) = \tilde{O}(n)
\]

Offsets for 1-bits in sparse chunks
Clark's select

So far, strategy for select is:

(a) find what chunk it's in (division)

(b) if chunk is sparse (≥ log⁴ n bits)
   (b.i) look up in sparse offset table

(c) if chunk is dense (< log⁴ n bits)

TODO

So far, space is Õ(n)
Clark's select

Split into $\log^2 n$-weight chunks

Sparse ($\geq \log^4 n$-length)

Lookup table for each 1-bit $\checkmark$

Sparse ($\geq \frac{1}{2} \log n$-length)

Lookup table for each 1-bit $\checkmark$

Dense ($< \log^4 n$-length)

Split into $\sqrt{\log n}$-weight sub-chunks

Dense ($< \frac{1}{2} \log n$-length)

Lookup table for all possible sub-chunks $\checkmark$
Clark's select: dense case

**Dense** chunks are shorter than $\log^4 n$ bits; further subdivide these to sub-chunks of $\sqrt{\log n}$ 1-bits each.
Clark's select: dense case

Store *relative offset* per sub-chunk

There are \( \leq n/\sqrt{\log n} \) sub-chunks

Since containing chunk has length

\(< \log^4 n\) bits, *relative offset* fits in

\(O(\log \log^4 n) = O(\log \log n)\) bits

Overall:

\[
O \left( \frac{n \log \log n}{\sqrt{\log n}} \right) = \tilde{o}(n)
\]
Clark's select

So far, strategy for select is:

(a) find what chunk it's in (division)
(b) if chunk is sparse
   (b.i) look up in sparse offset table
(c) if chunk is dense
   (c.i) look up chunk’s offset
   (c.ii) find what sub-chunk it’s in (division by $\sqrt{\log n}$)
   (c.iii) look up sub-chunk’s relative offset

TODO: need to look within sub-chunks
Clark's select: dense/sparse case

Sub-chunks with $\geq 1/2 \log n$ bits are **sparse**; we simply store relative offsets for every 1-bit.

Overall: $O\left(\frac{n}{1/2 \log n}, \log \log n, \sqrt{\log n}\right)$

Max # sparse sub-chunks

# bits to store

1 answer (rel. to chunk)

# 1-bits per chunk

= $O\left(\frac{n \sqrt{\log n \log \log n}}{\log n}\right) = O\left(\frac{n \log \log n}{\sqrt{\log n}}\right) = \tilde{o}(n)$
Clark's select

Split into $\log^2 n$-weight chunks

Sparse ($\geq \log^4 n$-length)

Lookup table for each 1-bit ✅

Dense ($< \log^4 n$-length)

Split into $\sqrt{\log n}$-weight sub-chunks

Sparse ($\geq 1/2 \log n$-length)

Lookup table for each 1-bit ✅

Dense ($< 1/2 \log n$-length)

Lookup table for all possible sub-chunks ✅
Clark's select: dense/dense case

Sub-chunks $< \frac{1}{2} \log n$ bits are dense; pre-calculate answers for all such chunks, like rank:

$$2^{1/2 \log n} \cdot \sqrt{\log n} \cdot \log \log n$$

possible bitvectors possible 1-bits answer

$$= O \left( \sqrt{n \log n \log \log n} \right)$$

$$= \tilde{o}(n)$$
Clark's select

(a) find what chunk it's in (division by $\log^2 n$)

(b) if chunk is **sparse** ($\geq \log^4 n$ bits)
   (b.i) look up answer in sparse offset table

(c) if chunk is **dense** ($< \log^4 n$ bits)
   (c.i) look up chunk’s offset
   (c.ii) find what sub-chunk it’s in (divide by $\sqrt{\log n}$)
   (c.iii) look up sub-chunk’s relative offset
   (c.iv) if sub-chunk is **sparse** ($\geq 1/2 \log n$ bits)
      (c.iv.A) look up answer in sparse 1-bit table
      (c.iv.B) return (c.i) + (c.iii) + (c.iv.A)
   (c.v) if sub-chunk is **dense**
      (c.v.A) look up answer in all possible dense/dense table
      (c.v.B) return (c.i) + (c.iii) + (c.v.A)
Clark's select

Overall, space is $\tilde{o}(n)$
## Bitvectors

<table>
<thead>
<tr>
<th></th>
<th>Time</th>
<th>Space (bits)</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>$B . \text{access}$</td>
<td>$O(1)$</td>
<td>$n$</td>
<td>Lookup</td>
</tr>
<tr>
<td>$B . \text{select}_1$</td>
<td>$O(1)$</td>
<td>$\tilde{o}(n)$</td>
<td>✓ Clark</td>
</tr>
<tr>
<td>$B . \text{rank}_1$</td>
<td>$O(1)$</td>
<td>$\tilde{o}(n)$</td>
<td>✓ Jacobson</td>
</tr>
</tbody>
</table>