LECTURE 7: WRITE-ONCE MEMORY (WOM) – CONSTRUCTIONS AND BOUNDS
WOM Quiz

\[
\psi(c_1, \ldots, c_n) = (y_1, \ldots, y_k)
\]

\[
w(\langle 2^k \rangle^t) \leq kt
\]

\[
w(\langle v \rangle^{t_1 + t_2}) \leq w(\langle v \rangle^{t_1}) + w(\langle v \rangle^{t_2})
\]

\[
\mu(c_1, \ldots, c_n ; y_1, \ldots, y_k) = (c'_1, \ldots, c'_n)
\]
The Linear WOM Code

\[ v, n=v-1, t=v/4 \]

Decoding function:

\[ x = \sum_{i=1}^{v-1} ic_i \mod v \]

Update function:

Current info: \( x \)
New info: \( y \)

Change \( c_i \)s \( 0 \rightarrow 1 \) such that

\[ y = \sum_{i=1}^{v-1} ic'_i \mod v \]

Minimally!
Linear WOM Code – Example

v=6, n=5

Decoding:

\[ x = (1 + 4) \mod 6 = 5 \]

Update:

New info: \( y = 0 \)

Need +1, but cell 1 already “1”

Update +7 instead

\[ (1 + 2 + 4 + 5) \mod 6 = 0 = y \]
Linear WOM Code - Update

Add \((y-x) \mod v\) to the array by "minimal" 0→1 changes. Define \(z=(y-x) \mod v\)

**Case 1:** \(c_z = 0\)  \hspace{1cm} \(c_z: 0 \rightarrow 1\)
Add \((y-x)\mod v\) to the array by “minimal” 0→1 changes.
Define \(z = (y-x)\mod v\)

**Case 2:** \(c_z = 1\)

Define \(S\) to be the set of zero bits: \(S = \{i: c_i = 0\}\)
Assumption: \(|S| \geq \frac{v}{2}\)
Define \(T\) as the set \(T = \{(z - i) \mod v \mid i \in S\}\)
Fact: \(|T| = |S| \geq \frac{v}{2}\)
Linear WOM Code – 2-bit Update

\[
z = (y-x) \mod v \quad S = \{i: c_i = 0\} \quad T = \{(z - i) \mod v | i \in S\}
\]

Fact 1: z not in S  
Fact 2: z not in T  
Fact 3: z not 0

Otherwise Case 1
There is no \(c_0\)
That would mean \(y = x\)

If \(|S|, |T| \geq \frac{v}{2}, |T \cup S| < v - 1\) then \(|T \cap S| \geq 2\)
Linear WOM Code – 2-bit Update

\[ z = (y - x) \mod v \]

\[ S = \{ i : c_i = 0 \} \]

\[ T = \{ (z - i) \mod v \mid i \in S \} \]

\[ |T \cap S| \geq 2 \]

1) Choose \( i' \) from the intersection \( T \cap S \)

2) \( i' = (z - i'') \mod v \), for some \( i'' \) in \( S \)

3) Both \( i', i'' \) in \( S \), and \( i' + i'' = z \mod v \)

Add \( z \) to the array by \( c_{i'}: 0 \rightarrow 1 \)

\( c_{i''}: 0 \rightarrow 1 \)

\( \leq 2 \) 0→1 changes for every write

Guaranteed \( v/4 \) writes!

\( \sqrt{\text{Done!}} \)
Getting More than n/4 Writes

What to do with the remaining \( v/2 \) cells still at zero?

Divide the block to \( n/3 \) triples:

\[
\begin{array}{cccccc}
110 & 011 & 100 & 111 & 010 \\
\end{array}
\]

Ex. \( n=15 \)

Rules:
1) If 0 “1”s: add arbitrary “1”
2) If 2 “1”s: add a “1” to get 111
3) Now re-use all **triples with a single “1”**

Single “1” means 0

Two “1”s means 1
Getting More Than n/4 Writes

<table>
<thead>
<tr>
<th>After v/4 writes</th>
<th>110</th>
<th>011</th>
<th>100</th>
<th>111</th>
<th>000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prepare next write</td>
<td>111</td>
<td>111</td>
<td>100</td>
<td>111</td>
<td>010</td>
</tr>
</tbody>
</table>
| After next write | 111 | 111 | 101 | 111 | 010 | x | x | 1 | x | 0 | Ex. n=15
How Many More Writes?

How many additional bits from triples?

n/3 triples, n/2 “1”s total.

Worst case: n/2 “1”s in n/4 triples with 2 “1”s

# additional bits $\geq n/3 - n/4 = n/12$

Recursion: $t(n) = n/4 + t(n/12)$

# writes total: $t(n) = 3n/11 + O(1)$ → Recursion improved t by factor 12/11
And now for something completely different...
The first write needs to leave at least $\log_2(v)$ zero bits

Define $m = \log_2(v)$ and $h = n - m$ bits for first write

Condition on $h$ from first write:

$$\sum_{i=0}^{h} \binom{n}{i} = \sum_{i=0}^{h} \binom{m + h}{i} \geq v$$

$$w(\langle v \rangle^2) \geq m + \min \left\{ h : \sum_{i=0}^{h} \binom{m + h}{i} \geq v \right\}, m = \log_2(v)$$
Lower Bound on $w(\langle v \rangle^t)$

$v, t, n \geq ?$

Define

$$\delta(v, m) = \min \left\{ h : \sum_{i=0}^{h} \binom{m}{i} \geq v \right\}$$

**Theorem:**

$$w(\langle v \rangle^t) \geq z(v, t)$$

where

$$z(v, 0) = 0$$

$$z(v, t + 1) = z(v, t) + \delta(v, z(v, t)) \quad t \geq 0$$

**Proof:** by induction

For first write

$$\delta(v, z(v, t))$$

Zeros for writes $2, \ldots, t+1$

$$z(v, t)$$
Explicit Lower Bound on $w(\langle v \rangle^t)$

$v=2^k, t, n \geq ?$

**Corollary:**

$w(\langle 2^k \rangle^t) \geq k + t - 1$

**Proof:**

$z(2^k, 1) = \delta(2^k, 0) = k$

$z(2^k, t + 1) \geq z(2^k, t) + 1$

Last write: k bits
Each earlier write, at least 1 additional bit
Explicit Lower Bound on $w(\langle v \rangle^2)$

$v, t=2, n \geq ?$

**Theorem:**

$$w(\langle v \rangle^2) \geq 1.293 \log_2(v)$$

**Proof:**

$$\sum_{i=0}^{n-\log_2(v)} \binom{n}{i} \geq v$$

The asymptotic wand

$$2^n H\left(\frac{n-\log_2(v)}{n}\right) \geq v$$

The binary entropy function

Then solve $H\left(1 - \frac{\log_2(v)}{n}\right) \geq \frac{\log_2(v)}{n}$ and get $n \geq 1.293 \log_2(v)$
**Definition:** WOM Capacity

Define the **t-write WOM capacity** $C_t$ as the total number of bits that can be written in all $t$ writes, normalized by the number of cells $n$.

For $t=2$,

$$w(\langle v \rangle^2) \geq 1.293 \log_2(v)$$

$$C_2 \leq \frac{2 \log_2(v)}{1.293 \log_2(v)} = 1.54$$

Are you sure?
Variable-Rate WOM

We can write $v_1$ in the first write and $v_2$ in the second write. Will it help?

<table>
<thead>
<tr>
<th>Bits for first write</th>
<th>Bits for second write</th>
</tr>
</thead>
<tbody>
<tr>
<td>$pn$</td>
<td>$(1 - p)n$</td>
</tr>
</tbody>
</table>

$v_2 = 2^{(1-p)n}$

$v_1 = \sum_{i=0}^{pn} \binom{n}{i} \approx 2^{nH(p)}$
Maximal Sum-Rate

Total number of bits:

\[ \log_2(v_1) + \log_2(v_2) = nH(p) + (1 - p)n \]

Maximize over \( p \):

\[ \bar{p} = \arg\max_p [H(p) + (1 - p)] = \frac{1}{3} \]

\[ C_2 \leq H(\bar{p}) + (1 - \bar{p}) = 1.58 > 1.54 \]

0.918 \hspace{1cm} 0.666