sketch (x_0) = extract bits $b_0, b_1, \ldots, b_{r-1}$ out of word x
i.e. r-bit vector, ith entry = b_ith bit of x

~~Problem: sketch is hard to compute~~

~~But~~ this would reduce node representation to $\underbrace{k}_{\Theta(B)}$ $\underbrace{r}_{O(B)}$-bit strings

$= O(B^2) = O((\lg n)^{2/5})$ bits total
< 1 word!

Problem: sketch is hard to compute. (esp. for query q)

Solution: compute a slightly less compact form of sketch(x).
Contains the same bits as vsketch, in same order,
but spread out with perfect extra 0^* bits in between.
(in pattern that's independent of x i.e. consistent)

To do this: use multiplication.

$$x \cdot m = \left(\sum_{i=0}^{r-1} x_{b_i} 2^{b_i}\right)\left(\sum_{i=0}^{r-1} 2^{m_i}\right)$$
$$= \sum_{i=0}^{r-1}\sum_{j=0}^{r-1} x_{b_i} 2^{b_i+m_j}$$

Our goal: pick m so that b_i+m_j are all distinct
& b_i+m_i are concentrated in small interval of bits

$\Rightarrow$ then can take $(x \cdot m)$ AND $\left(\sum_{i=0}^{r-1} 2^{b_i+m_i}\right)$
$\gg \min_{0\le i<r}\{b_i+m_i\}$
& get $\left(\max_{0\le i<r}\{b_i+m_i\} - \min_{0\le i<r}\{b_i+m_i\}\right)$-bit sketch.

First: pick $m_0', m_1', \ldots, m_{r-1}'$ so that b_i+m_j are distinct mod r^3.
- pick first to be 0 by induction
- m_i' must avoid $m_j' + b_j - b_k$ $\forall i,j,k$

$< r^3$ of them

$\Rightarrow m_i'$ can be picked without conflict.

Second: add suitable multiples of r^3 to $m_i' \hookrightarrow m_i$.
so that $b_0+m_0 < b_1+m_1 < \cdots < b_{r-1}+m_{r-1}$
& all fall just beyond highest bit of low half of product.