6.897
L12.2
Apr. 7, 2003

Rank & Select in a static bitstring

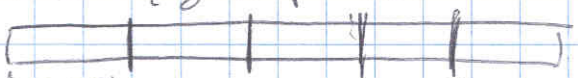
Rank(i) = #1's at or before index i
Select(i) = index of ith 1 bit.

⇒ left-child(i) = 2 rank(i)
right-child(i) = 2 rank(i) + 1
parent(i) = select(i div 2).

Goal: O(1) time query, o(n) extra space.

Rank [Jacobson 1989] (PhD thesis)

- can use lookup table for bitstrings of length $\frac{1}{2}\lg n$
  $O(\sqrt{n}\lg n \lg(\lg n))$ space
- but if we split into $2n/\lg n$ pieces, takes $\Theta(n)$ space
- instead split into $n/\lg^2 n$ pieces:



storing these (cumulative ranks) costs $O(\frac{n}{\lg^2 n}\lg n)$
$= O(n/\lg n)$ bits.

- split $(\lg^2 n)$-size chunks into $(\frac{1}{2}\lg n)$-size subchunks

storing accumulative ranks within chunks
only $O(\frac{n}{\lg n}\lg\lg n)$ bits.

- at bottom level, use lookup table.

Select also possible [Clark & Munro – PhD thesis 1996]
but much harder