6.897
L22.2
Apr. 28, 2005

even-succ$_k$ representation

- trivial for $SA_k[i]$ even (testable via bitvector)
- store answers for all $n_k/2$ i's for which $SA_k[i]$ odd
- order by i i.e. by order of these odd suffixes $T_k[SA_k[i]:]$

$\Rightarrow$ even-succ$_k(i)$ = $^{\text{odd-}}$answers[odd-rank(i)]
= i-even-rank(i)

= order by $(T_k[SA_k[i]], \underbrace{T_k[SA_k[i]+1:]}_{\text{even suffix}})$

$= (T_k[SA_k[i]], T_k[SA_k[\text{even-succ}(i)]:])$

= order by $(T_k[SA_k[i]],$ even-succ$(i))$ (property of SA_k)

$\Rightarrow$ store $T_k[SA_k[i]]$ prefix before each answer even-succ(i), (ordered by i),
& then stored values are themselves sorted

- to store $n_k/2$ sorted values of $2^k + \lg n_k$ bits each, subject to array indexing in $O(1)$:
  - split off leading $\lg n_k$ bits from each value v_i
  - store leading bits by unary differential encoding:
    $0^{\text{leading}(v_0)}\,1\,0^{\text{leading}(v_1)-\text{leading}(v_0)}\,1\,0^{\text{leading}(v_2)-\text{leading}(v_1)}\,1 \ldots$
    $\Rightarrow n_k/2$ 1's & $\le 2^{\lg n_k} = n_k$ 0's
    $\Rightarrow \le \frac{3}{2} n_k$ bits
  - store trailing 2^k bits of each v_i explicitly in order
    $\Rightarrow (n_k/2) \cdot 2^k = n/2$ bits
  - random access / array indexing:
    - random access into trailing bits (fixed sizes)
    - use select$_1$ structure to find leading bits:
      leading(v_i) = select$_1(i)$ − select$_1(i-1)$ − 1

- space: $\frac{1}{2}n + \frac{3}{2}n_k + O(n_k / \lg \lg n_k)$

$\Rightarrow$ total space for even-succ $= \frac{1}{2} n \lg \lg n + 3n + O(\frac{n}{\lg \lg n})$

$\Rightarrow$ total space for $O(\lg \lg n)$ structure $= \frac{1}{2} n \lg \lg n + 5n + O(\frac{n}{\lg \lg n})$