Succinct data structures (usually static)

6.897
L12.1
Apr. 7, 2003

Goal: space = information-theoretic optimum + o(that)
& still able to manipulate structure as before

Binary trees/tries (labeled edges)

- How many rooted labeled binary tries are there on n nodes?
  Catalan number $C_n = \binom{2n}{n}/(n+1)$
- How many bits does it take to encode such a trie? (information theory)

$$\lg C_n = \lg \frac{(2n)!}{(n!)^2} - \lg(n+1)$$

$$= \lg(2n)! - 2\lg n! - \lg(n+1)$$

$$= (2n)\lg(2n) - 2n\lg n + o(n) \quad \text{- Sterling}$$

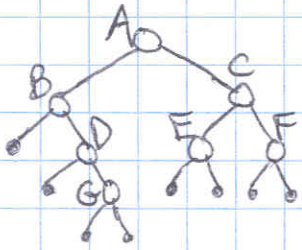
$$= (2n)(1+\lg n) - 2n\lg n + o(n)$$

$$= 2n + o(n).$$

Level-order representation

- Append external node for each missing child (already there in compacted trie)
- Read nodes in level order
- Write 1 for internal node, 0 for external node

Ex:



1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A	B	C	·	D	E	F	·	G	·	·	·	·	·	·
1	1	1	0	1	1	1	0	1	0	0	0	0	0	0

- 2n+1 bits — perfect
- can we still manipulate it?
  — to search a la trie, need left child/right child

Claim: positions of left & right children of ith internal node are 2i and 2i+1.

Proof: Suppose there are j external nodes up to ith internal node. Then strictly between ith internal node & its left child there are i−j−1 nodes

- offspring of i−1 previous internal nodes (2(i−1))
- i−1 of which were internal nodes (all but root)
- j of which were external nodes

∴ position of left child = $i + j + (i - j - 1) + 1 = 2i$. □