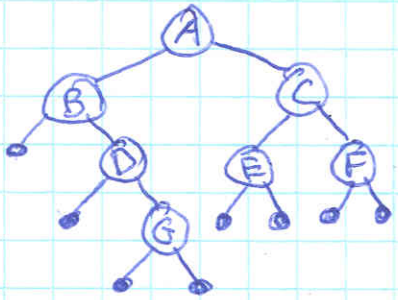
Level-order representation of binary tree

6.897
L 21.2
Apr. 26, 2005

- append external node for each missing child
- read nodes in level order
- write 1 for internal node, 0 for external node
- equivalently: internal nodes write {have left child?} {have right child?}

e.g.



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 ~ great
- positions of left & right children of ith internal node are $2i$ & $2i+1$.
  - suppose ith internal node at position $i+j$ i.e. j external nodes up to ith internal node
  - $i-1$ previous internal nodes have $2(i-1)$ children
  - $i-1$ already seen as internal (all but root)
  - j already seen as external

$\Rightarrow$ left child at position $(i+j) + \underbrace{2(i-1) - (i-1) - j}_{\#\text{ intervening nodes}} + 1 = 2i$

Rank & select in bitstring

- $\text{rank}_1(i)$ = # 1's at or before index i
- $\text{select}_1(i)$ = index of ith 1 bit

$\Rightarrow$ left-child(i) = 2 rank(i)
right-child(i) = 2 rank$(i)+1$
parent(i) = select$(\lfloor i/2 \rfloor)$