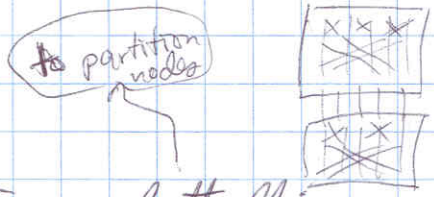
6.897
L19.4
May 5, 2003

Recovery time:

- $O(f \lg f)$ inaccessible nodes $\Rightarrow O(f \lg f)$ layers considered
  $\Rightarrow O(sf \lg f)$ time
- In fact, $\leq f$ layers with inaccessible nodes before accessible layer hit.
  $\Rightarrow O(s \cdot f)$ time



Linked list

- same underlying structure as stack except add one purely vertical layer between butterflies
- Invariant: each butterfly (except top) stores $\Theta(s \lg s)$ nodes from linked list & these nodes are consecutive in the list
- within a butterfly, however, nodes are in arbitrary "order" (next pointers are no longer horizontal)
- each butterfly stores freelist of remaining nodes — so really 2 linked lists in each butterfly — & only 1 links on to the next butterfly.
- Insert & Delete operate on butterfly of specified node & use B-tree split/merge trick to maintain invariant
- recovery uses order-queries-on-list DS in each butterfly to restore order despite arbitrarily lost nodes
  - $O(\lg s)$ list labeling $\Rightarrow O(\lg s)$ overhead in Insert/Delete
  - $O(1)$ via indirection $\Rightarrow O(\lg s)$ overhead in # lost nodes (may lose entire block of $\Theta(\lg s)$ per original loss)

Further: [redacted] — trees — above/below & left/right relations
— expander graphs shave $\alpha(\lg f)$ factor

OPEN: — RAM version (never lose anything)
— any pointer-machine DS?