Fault-tolerant data structures [Aumann & Bender-FOCS1996]

6.897
L19.1
May 5, 2003

Example: linked list

one fault ⇒ entire list is lost!

Model:

- pointer machine
- at arbitrary times, node can go faulty ⇒ all its data lost
- faulty nodes detected upon access (e.g. error-detection codes)
- unlimited supply of "replacement nodes"
- s words of safe/secure memory
  - connection between DS and outside world
  - without this assumption, can easily lose entire DS
  - in reality, registers etc.
    (smaller levels of memory hierarchy easier to make fault-free)

Goals:

- build fault-tolerant versions of standard data structures
- behavior of normal & FT DS's should be the same
- running times should be same up to constant factors
- if more than $c \cdot s$ faults for some c, FT DS breaks
  (if $c=1$, this is the best we could hope for)
- if $f \le cs$ faults simultaneously (before fixing any),
  - lose $O(\ell(f))$ nodes — independent of n
  - recover working DS in $O(r(f,s))$ time
  - recovered elements preserve desired properties
    e.g. relative order