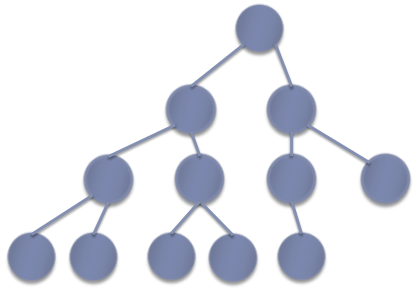
cs140 – algorithms
prof. yi chen

september 19, 2013

Analyzing data structures

- How efficiently does it execute supported operations?
- On *average* how efficiently does it execute supported operations?



Amortized analysis

- idea: analyze a sequence of operations in order to show that although a single operation may be expensive, on average the cost per operation is small.
- observation: *not* probabilistic analysis
- plan: go over 3 methods for doing amortized analysis

Multipop stacks

- Operations
  - push(S,d)
  - pop(S)
  - multipop(S,k)
- Implementation?
- Running time for each operations?
- Running time for sequence of d operations?

Aggregate analysis

- Idea:
  - Derive an upper bound T(d) on the cost of any sequence of d operations.
  - Use T(d)/d as the average cost of any operation on the data structure.
- Application to multipop stacks

Accounting method

- Idea:
  - Define amortization scheme
  - Justify differences between actual and amortized cost for each operation (i.e. why the scheme leaves you debt-free)
- Application to multipop stacks

Potential method

- Idea:
  - Define: potential function ϕ which maps state of data structure to a real number
  - Justify: why potential function never goes below initial value
  - Amortized cost for operation is actual cost + $\Delta\,\phi$
- Application to multipop stacks

Amortized analysis of binary counters

- increment(B)

0	0	0	0	0	0
0	0	0	0	0	1
0	0	0	0	1	0
0	0	0	0	1	1
0	0	0	1	0	0
0	0	0	1	0	1
0	0	0	1	1	0
0	0	0	1	1	1
0	0	1	0	0	0

Amortized analysis of dynamic tables

- insert(T,x)
- delete(T,x)