LEC 17

CSE 123

Hashing

BEFORE WE START

Talk to your neighbors:

Do you usually remember where you put things?

Instructor: James Wilcox

Questions during Class?

Raise hand or send here

sli.do #cse123

Announcements

- P3 due tonight!
- R7 due tonight

- There will be an extra resub open to any assignment.

Data structures so far

- **Lists**
  - Maintain an ordered sequence of elements
  - Provides get(), add(), remove(), ...
  - Studied two implementations: ArrayIntList and LinkedIntList

- **Sets**
  - Maintain a collection of elements
  - Provides contains(), add(), remove(), ...
  - Implementations?

Set implementations

	ArraySet(?)	LinkedSet(?)	TreeSet	HashSet
contains()	O(n)	O(n)		
add()	O(n)	O(n)		
remove()	O(n)	O(n)		

Set implementations

	ArraySet(?)	LinkedSet(?)	TreeSet	HashSet
contains()	O(n)	O(n)	O(log(n))*	
add()	O(n)	O(n)	O(log(n))*	
remove()	O(n)	O(n)	O(log(n))*	

* assuming tree is balanced

Set implementations

	ArraySet(?)	LinkedSet(?)	TreeSet	HashSet
contains()	O(n)	O(n)	O(log(n))*	O(1)**
add()	O(n)	O(n)	O(log(n))*	O(1)**
remove()	O(n)	O(n)	O(log(n))*	O(1)**

* assuming tree is balanced

** assuming collisions are not too bad

Hash Table

- Assume we have a hash function for the elements
  - Takes an element and returns an integer
- Make an array of N “slots”, all initially empty
- Idea: put each element x at the index h(x) mod N