Direct Addressing + Open Addressing

Data Structures of Set ADT

next class: Separate Chaining

Suppose we have a set of
n area codes that we
want to maintain

area code, for example, could be an
instance var in a "County" object
Direct Addressing

insert, locate, remove

000
001

314

Object for area code 314
(bucket/collection of all countries with this area code)

999
array

called a table
Key here

every element you might possibly insert into set has a dedicated index

slot in the table

worst-case

insert - table[slot] = element \( \Theta (1) \)

locate - access table [slot] \( \Theta (1) \)

remove - table [slot] = null EMPTY \( \Theta (1) \)
Let's make this a little more general

equivalence tester
define how check for equivalence

hasher

hashCode mapping from object to some integer in \{0, 1, \ldots, m-1\}
What is the big limitation?

Size of table is as big as universe of all possible elements that might be inserted

Consider a uni of 5000 students where SS # is the id #.

$10^9$
Direct addressing is only a reasonable choice (in terms of space usage) when roughly \( n > \frac{1U1}{4} \)

\# elements held in set