Week 10: Maps and Binary Search

Thursday, November 12, 2020 9:46 AM

- keys → generic → unique
- values → generic

Operations

- get(Object)
  - returns value associated with key
  - if key not in map: returns null
- put(k, v)
  - if map does not have key:
    - adds key & value
    - null
  - else:
    - updates value
    - returns old value
- clear, size,
- remove(Object)

→ remove (Object)
  → removes key & value
  → returns value associated w/ key
→ containsKey, containsValue

Keys	Values
Earth	Planet
Venus	Planet
Sirius	Star
Andromeda	Galaxy
Milky Way	Galaxy
Europa	Moon

Different Views

→ entrySet
  → every element is unique

- every element is unique in a Set
- key/value pairs: "entries"

<Earth, Planet>, <Europa, Moon>

Map.entrySet()
- use iterator/enhanced for loop to access
- operations:
  - contains(Object)
  - remove(Object)
  - iterator, clear, equals

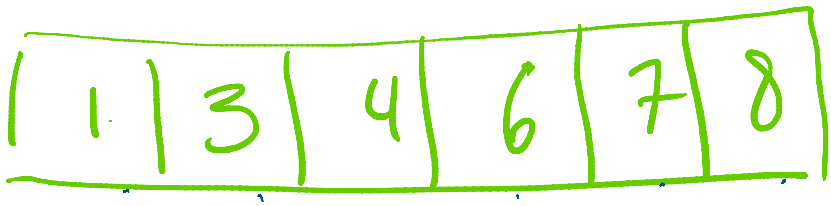
- keySet
  - Set of keys
  - operations:
    - Same as entrySet

- values

→ Values
→ Collection
→ values can have duplicates

* all using same data structure as map
* don't have to implement keySet or Values
→ use entrySet

Binary Search



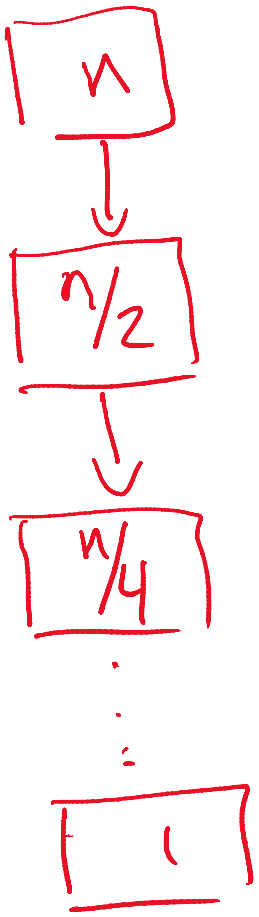
→ search for 5
→ simplest: linear search

O(n)

→better: binary search

→search in BST:

→go left if element is smaller, go right if element is bigger than current node



*O(log n)

→cut search space in half each time

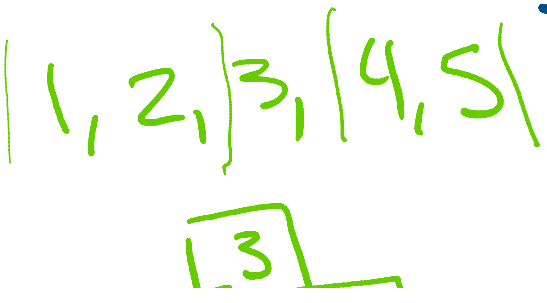
→ with an array:

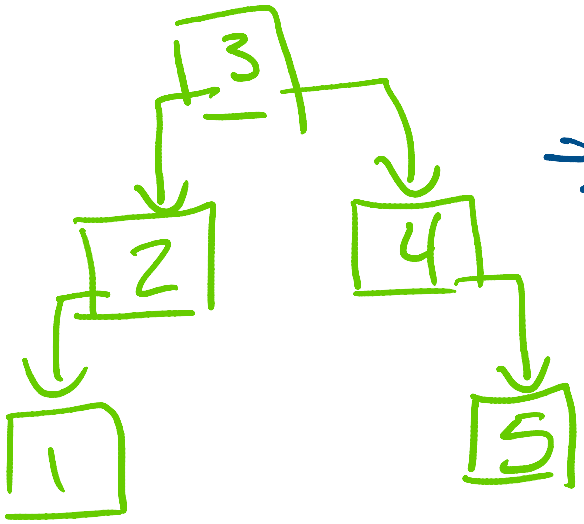
→lower bound

→inclusive - part of search space

→upper bound

→exclusive - not part of search space





'space'

→ find midpoint & compare to the element you are looking for