# Hash tables

- Want a way to store things and look them up quickly
- Naive way: look through the whole array: A = [2, 1, 3, n, n-1, ..., 4]
- How long to check presence?
  - O(n): may need to check every element in order
- Can do better if array is sorted:
  - Binary search: O(log(n))
- What about if data is unsorted?

# Observations about lookup speed

- We know that array lookups for a given index are O(1)
- Idea: we can store elements in an array, at their own index
- A = [1, 2, 3, 4, 5, 6, ..., n]
- Then, we can check if an element is there by checking that index in O(1) time
- Downside: space complexity will need to be O(max(A) - min(A)) to store elements uniquely, and will need to resize array whenever larger element is added.
  - Example: A = [1, 1,000,000] will require 1,000,000 slots to store 2 items