CS 240

Data Structures and Algorithms I

Alex Vondrak

ajvondrak@csupomona.edu

November 23, 2011

Searching

Linear

```
boolean search(int needle, int[] haystack) {
    for(int item : haystack) {
        if(item == needle) return true;
    }
    return false;
}
```

```
boolean search(int needle, int[] haystack) {
   return search(needle,
                 haystack,
                 0,
                 haystack.length-1);
}
```

Searching
Binary Search

```
private boolean search(int needle, int[] haystack,
                       int l, int r) {
   if (l > r) return false;

   int mid = (l+r)/2;

   if (needle == haystack[mid]) return true;

   if (needle < haystack[mid])
      return search(needle, haystack, l, mid-1);

   if (needle > haystack[mid])
      return search(needle, haystack, mid+1, r);
}
```

Searching

Analysis

- Of course, linear search is $O(n)$ in the worst case
- Denote the worst-case running time of a binary search $T(n)$, where $n = r - l$

$$\begin{aligned} T(1) &= c_1 && \text{(where } c_1 = \text{some constant)} \\ T(n) &= c_2 + \underbrace{T(n/2)}_{\text{recursive call}} \end{aligned}$$

$T \in O(\log_2 n)$, since $\log_2 n = p$ such that $2^p = n$—i.e., it's how many times we can (destructively) divide n by 2 until the quotient reaches 1

Searching
Hashing

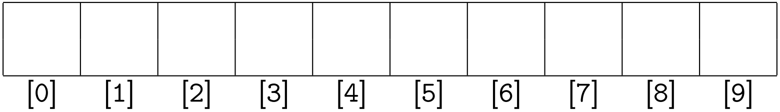
Idea: design a data structure in such a way that we know where any particular element should be stored

Example

Suppose we have the following data:

38 16 47 15 53 90 29

How we we store it in an array of length 10? What happens when we search for, say, 48?



Searching
Hashing

Idea: design a data structure in such a way that we know where any particular element should be stored

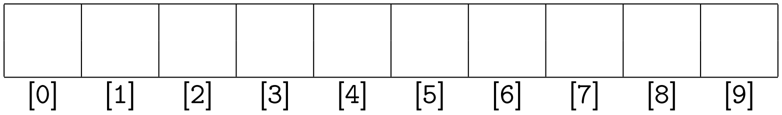
Example

Suppose we have the following data:

38 16 47 15 53 90 29

How we we store it in an array of length 10? What happens when we search for, say, 48?

- Hash function: `data[hash(i)] = i`; let's try `hash(i) = i % 10`



Hash Tables

- Take the idea of a hash function storing objects in an array...
- ...But use two distinct parameters

Before

```
data[hash(i)] = i;
```

After

```
data[hash(k)] = v;
```