Hashtables

Lunch & Learn

Ethan Kent

January 30, 2020

Purpose

The goal of this presentation is to discuss what hashtables are, how they work, and the tradeoffs that go along with different designs.

Data structures and tradeoffs

Linked lists

`O(1)` insertion and popping; `O(n)` for search, access by position.

```
Node = Struct.new(:value, :next_node)

node1 =  Node.new(0)
node2 =  Node.new(1)
node3 =  Node.new(2)
node4 =  Node.new(3)

node1.next_node = node2
node2.next_node = node3
node3.next_node = node4
node4.next_node = :null_pointer

puts node1
```

Data structures and tradeoffs

Arrays

Linked lists have pointers to arbitrary other locations in memory. By contrast, arrays are contiguous locations in memory.

Arrays have `O(1)` insertion, popping, and access by position, and `O(n)` search.

```
int * my_array;

my_array = (int * ) malloc(sizeof(int) * 50);

for(i = 0; i < 50; ++i) {
    my_array[i] = 0;
}
```

Data structures and tradeoffs

The desire for constant-time "find" operations

- With a linked list, we can only find an element by starting at the beginning and searching every spot. We can't even go directly to an element if we know its index.
- With a vector, we can only find an element by starting at the beginning and searching every spot. But we *can* go directly to an element if we know its index.

It would be nice would be to have some way to use the value we're checking for *to tell us where to look*, so we could go directly there.

An analogy

What if we had a machine that could use the description of your backpack:

and tell you it belongs in locker 57?

An analogy

What else would our machine have to do? It should—

- Always give you the same answer given the same backpack.
- Not put every backpack in locker 57.
- Not put most backpacks in just a few lockers.
- Not put most blue backpacks in just a few lockers.
- Give you an answer pretty quickly.

If we had that, we could walk directly do locker 57 and get our backpack, without the need to search.

An analogy

But there are still some problems. What if—

- More than one backpack is assigned to the same locker (assume only one backpack can fit)?
- There are more backpacks than lockers?

Hashtables

We have now considered several aspects of a data structure called a *hashtable*.

- The machine that turns the backpack into a locker assignment is hash function.
- The characteristics of a good backpack assigner are the characteristics of a good hash function:
  - Deterministic,
  - Fast,
  - Uniform, and
  - Avalanching.
- The problems you can run into are the same as those with a hashtable:
  - Collisions, and
  - Load factor.