lecture22: Hashing

Sean Massung

Largely based on slides by Cinda Heeren
CS 225 UIUC

17th July, 2013
Announcements

- lab_avl due tomorrow (7/18)
- mp6.1 extra credit due Friday night (7/19)
- lab_heaps out tomorrow, due Saturday night (7/20)
Say you have a list of names, and you want to figure out which area is associated with each name. Can you use an efficient data structure to do this?

<table>
<thead>
<tr>
<th>Name</th>
<th>Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dan Roth</td>
<td>Machine Learning and NLP</td>
</tr>
<tr>
<td>Chengxiang Zhai</td>
<td>Information Retrieval</td>
</tr>
<tr>
<td>Jiawei Han</td>
<td>Data Mining</td>
</tr>
<tr>
<td>Steve LaValle</td>
<td>Algorithms and Robotics</td>
</tr>
<tr>
<td>Jeff Erickson</td>
<td>Computational Geometry</td>
</tr>
<tr>
<td>David Forsyth</td>
<td>Computer Vision</td>
</tr>
</tbody>
</table>
Let me guess, you used a Dictionary ADT and can support insert, find, and remove in $O(\log n)$ time?

That’s pretty good, but what if you could do it in constant time? Look at the data again, and try to find some sneaky, “cheating” way to look up an area given a name.

<table>
<thead>
<tr>
<th>Name</th>
<th>Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dan Roth</td>
<td>Machine Learning and NLP</td>
</tr>
<tr>
<td>Chengxiang Zhai</td>
<td>Information Retrieval</td>
</tr>
<tr>
<td>Jiawei Han</td>
<td>Data Mining</td>
</tr>
<tr>
<td>Steve LaValle</td>
<td>Algorithms and Robotics</td>
</tr>
<tr>
<td>Jeff Erickson</td>
<td>Computational Geometry</td>
</tr>
<tr>
<td>David Forsyth</td>
<td>Computer Vision</td>
</tr>
</tbody>
</table>

How about a table of size 26?

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>(A)</td>
</tr>
<tr>
<td>1</td>
<td>(B)</td>
</tr>
<tr>
<td></td>
<td>...</td>
</tr>
<tr>
<td>5</td>
<td>(E)</td>
</tr>
<tr>
<td>6</td>
<td>(F)</td>
</tr>
<tr>
<td>7</td>
<td>(G)</td>
</tr>
<tr>
<td>8</td>
<td>(H)</td>
</tr>
<tr>
<td>9</td>
<td>(I)</td>
</tr>
<tr>
<td></td>
<td>...</td>
</tr>
</tbody>
</table>
A perfect hash function

- You just wrote a perfect hash function!
- \( \text{hash}(\text{name}) = \text{first char of last name } - 'A' \)
- A perfect hash function is a bijection between the keyspace and a collection of small integers
  - Each key maps to a different int (one-to-one)
  - Given 26 different last names, each name hashes on the entire sequence of ints (onto)
- Well, that was contrived, but can we use this general idea to create a fast implementation of the dictionary ADT?
  - Yes, of course!
Collisions

- Given a dataset \{200, 205, 210, 215, 220, ..., 595, 600\}, and a table of size \( N = 100 \), create a hash function
  - How about \( h(k) = k \mod N \)?
- A collision is when two different keys hash to the same location in the table
  - That is, \( k_1 \neq k_2, h(k_1) = h(k_2) \)
- Is our table too small? Well, there are 80 keys and 100 cells...
  - Better hash function: \( h(k) = \frac{k-200}{5} \)
- How will this work for other datasets?
Hash functions...

- Consist of two parts
  1. A hash: function mapping a key \( k \) to an integer \( i \)
  2. A compression: function mapping \( i \) onto array cells 0 to \( N - 1 \)

- SUHA: Simple Uniform Hashing Assumption
  - \( \forall k_1, k_2 : P(h(k_1) = h(k_2)) = \frac{1}{N} \)

- Choosing a hash function is tricky
  - Don’t create your own!
  - Smart people make dumb hash functions (Knuth’s multiplicative hash)

- What is a bad hash function?
  - Lots of collisions
  - Non-constant computation time
  - Nondeterministic
Good hash functions

Based on the previous list, good hash functions should satisfy the following:

1. Computed in $O(1)$ time
2. Deterministic: $k_1 = k_2 \rightarrow h(k_1) = h(k_2)$
3. Satisfy the simple uniform hashing assumption (SUHA)
Making a hash function

Say we want to map phone numbers (xxx-xxx-xxxx) to names.

Consider the following choices for hashing phone numbers:

- First 3 digits mod $N$
- Last 3 digits mod $N$
- Sort the numbers, then take the first 3 digits mod $N$
- Randomly select 3 digits, combine, and mod $N$
Hashing an object

We have the following Point class:

```c
struct Point {
    int x;
    int y;
    int z;
};
```

Consider the following hash functions for hashing a Point $P$:

- $h(P) = P.x \ % \ N$
- $h(P) = (P.x + P.y + P.z) \ % \ N$
- $h(P) = \max(P.x, P.y, P.z) \ % \ N$
- $h(P) = (31^0 \cdot P.x + 31^1 \cdot P.y + 31^2 \cdot P.z) \ % \ N$
As we’ve mentioned before, the C++ Standard Template Library (STL) has a wide array of data structures and algorithms. It has a tree-based Dictionary (\texttt{std::map}) that is implemented with a Red-Black tree. It also has a hash table (\texttt{std::unordered_map}). (Why is it called this?)

- \texttt{std::unordered_map} already has hash functions for some basic types like \texttt{int}, \texttt{std::string}, and \texttt{double}
- If you want to include your own objects in a \texttt{std::unordered_map}, you have to write your own implementation of a templated \texttt{hash<T>} function that returns a \texttt{size_t}
Like C++, Java has its own tree-based dictionary (java.util.TreeMap) using Red-Black trees. It also has its own hash table, java.util.HashMap, which you have probably heard of.

- All Java objects inherit a `hashCode()` function, which returns an int. Like C++, this is already implemented for some basic types
- You need to implement this function if your object is to be used in a HashMap or similar structure!
Hash Tables

A hash table is a Dictionary ADT implementation that makes use of hash functions to map (key, value) pairs to slots in an array. A hash table consists of:

1. An array  ✓
2. A hash function  ✓
3. A collision resolution strategy  (?)
Hash tables are useful because they offer amortized $O(1)$ running times for the dictionary operations. This running time comes from SUHA, as we will investigate next lecture.

Let’s think of some applications of hash tables (or dictionaries in general). If using a hash table, what hash function would you use?

- How can you efficiently store a large, sparse matrix?
- How can you keep track of whether a certain IP has visited a site before?
- How can you make a histogram of words from a text file?