CSCI 2270 Lecture Notes

3/6/19

Hashing algorithms continued:

Bias in Hashing

* for example, h(k) only returns even indices
* want to avoid

Multiplication Method

* + choosing m (table size), less likely to cause a bias
  + multiplication is less computationally expensive than modulo
  + less bias and less collisions than modulo method

1. Calculate sum of all ASCII values in key
2. Multiply by a constant A, where A is somewhere between 0-1.
   1. A can be derived heuristically
      1. heuristically means experimentation. Based on historical data, patterns, etc. Not a direct find of some sort.
   2. Common value for A = 13/32
3. Keep the fractional part of result
   1. this is the decimal part, take out the whole number part
4. Scale kA fractional by the table size (m)
   1. multiply kA by the table size
5. Take the floor of the result from step 4 to get integer index

Coding Q

* How do you get the fractional portion of a value?
  + integer = floor(value)
  + return value – integer

Perfect Hash Function

* assign values to array with no collisions and no wasted space

Ways to avoid collisions

* increasing table size above the possible range will not necessarily decrease the number of possible collisions
* pick table size such that it is the range of hashes

Collision Resolution

* one simple method is called open addressing
  + add new record at the next available location
* chaining
  + every array element is a pointer to the head of a linked list