CSI 3334, Data Structures

Lecture 12: Hash Tables

Date: 2012-10-10
Author(s): Philip Spencer, Chris Martin
Lecturer: Fredrik Niemelä

This lecture covers hashing, hash functions, collisions and how to deal with them, and probability of error in bloom filters.

1 Introduction to Hash Tables

Hash Functions take input and randomly sort the input and evenly distribute it.
An example of a hash function is randu:
$n_{i+1} = 65539 * n_i mod 2^{31}$
randu has flaws in that if you give it an odd number, it will only return odd numbers. Also, all numbers generated by randu fit in 15 planes.
An example of a bad hash function:
return 9

2 Dealing with Collisions

Definition 2.1 *Separate Chaining - Adds to a linked list when collisions occur.*

Positives of Separate Chaining:
There is no upper-bound on size of table and it's easy to implement.
Negatives of Separate Chaining:
It gets slower as you have more collisions.

Definition 2.2 *Linear Probing - If the space is full, take the next space.*

Negatives of Linear Probing:
Things linearly gather making it likely to cluster and slow down. This is known as primary clustering. This also increases the likelihood of new things being added to the cluster. It breaks when the hash table is full.

Definition 2.3 *Quadratic Probing - If the space is full, move i^2 spaces.*

Positives of Quadratic Probing:
Avoids the primary clustering from linear probing.

Negatives of Quadratic Probing:
It breaks when the hash table is full. Secondary clustering occurs when things follow the same quadratic path slowing down hashing.

Definition 2.4 *Double Hashing - Where there are two hashing functions.*

Positives of Double Hashing:
More random and avoids clustering.

Negatives of Double Hashing:
Breaks when it is full.

Definition 2.5 *Lazy Deletion - Places a marker where the deleted object was to tell the search function in linear probing, quadratic probing, and double hashing to continue searching.*

3 bloom filter

At the end of class, homework was assigned over the following problem:

There is a hash table that will contain every correctly spelled English word. This hash table is used to check if words are spelled correctly.
This hash table uses a bloom filter (several hash functions) to store 1's at hashes where correctly spelled words are located. What is the probability that an incorrectly spelled word will pass as being spelled correctly.
Use:
n = number of words
m = number of available spaces in table
k = number of hash functions