CSI33 Data Structures

Sharon Persinger

Fall 2019

Day 26 December 4

Topics

Hashing

Managing Collisions

- chaining
- open addressing

Hash Tables and hash functions

Collisions in hash functions

No matter what hash function we use, we will end up with two different keys that map to the same index.

With specific hash function, we can determine the keys.

Find two keys that are mapped to the same index by hash

```
def hash(key, array_size):

#key is an ASCI string
  coeffs = [2, 3, 5, 7, 11, 13]
  pos = 0

  for i in range(min(len(key), 6)):

            pos += coeffs[i] * ord(key[i])

  return pos % self.array_size
```

Dealing with collisions

Chaining:

- Store multiple (key, value) pairs in a list at the same index
- Easy to do on Python, since lists are heterogeneous
- Worst case – all key value pairs are stored at the same index location
- Must store both key and value at the index, so we can distinguish different pairs

- Example: HashTable

HashTable class – uses chaining

HashTable object has

Member variables:

self.object a list to store the (key, value) pairs

self.array_size, the size of the list

self.size, the number of items

self.coef, an array used in computing the hash function

Member methods:

Constructor

_hash function that computes

_setitem__ function to set the (key, value) pair by index. Look for the chaining.

__getitem__ function to retrieve the (key, value) pair by index. Again, look for chaining.

Dealing with collisions

Open addressing:

- Create some second function for finding a new index to use for the (key, value) pair if the index created by the hash function is already used for another pair
- Must store both key and value at the index
- Easiest approach is to use the next index that is not already being used – linear probing.

HashTable2 class – uses linear probing

HashTable2 object has

Member variables:

self.object a list to store the (key, value) pairs

self.array_size, the size of the list

self.size, the number of items

self.coef, an array used in computing the hash function

Member methods:

Constructor

_hash function that computes index for key

_setitem__ function to set the (key, value) pair by index. Look for the linear probing.

__getitem__ function to retrieve the (key, value) pair by index. Look for linear probing.

Write a hash function in C++

Use the same approach as in the HashTable examples.