CS 225

#27: Hashing, Part 2

October 30, 2017

Dictionary ADT in Client Code	
1	`Dictionary<KeyType, ValueType> d;`
2	`d[k] = v;`

A hash table consists of three things:

1.

2.

3.

Collision Handling Strategy #1: Separate Chaining

Example: **S = { 16, 8, 4, 13, 29, 11, 22 }, |S| = n**
h(k) = k % 7, |Array| = N

[0]	
[1]	
[2]	
[3]	
[4]	
[5]	
[6]	
[7]	

Load Factor:

Running time of Separate Chaining:

	Worst Case	SUHA
Insert		
Remove/Find		

Collision Handling Strategy #2: Probe-based Hashing

Example: **S = { 16, 8, 4, 13, 29, 11, 22 }, |S| = n**
h(k) = k % 7, |Array| = N

[0]	
[1]	
[2]	
[3]	
[4]	
[5]	
[6]	
[7]	

Linear Probing:
Try h(k) = (k + 0) % 7, if full...
Try h(k) = (k + 1) % 7, if full...
Try h(k) = (k + 2) % 7, if full...
...

Linear Probing leads to Primary Clustering

Description:

Remedy:

Double Hashing:

Example: **S = { 16, 8, 4, 13, 29, 11, 22 }, |S| = n**
$h_1(k) = k \% 7$, $h_2(k) = (5 - k) \% 5$, |Array| = N

[0]	
[1]	
[2]	
[3]	
[4]	
[5]	
[6]	
[7]	

Double Hashing:
Try $h(k) = (k + + 0*h_2(k)) \% 7$, if full…
Try $h(k) = (k + + 1*h_2(k)) \% 7$, if full…
Try $h(k) = (k + + 2*h_2(k)) \% 7$, if full…
…

$h(k, i) = (h_1(k) + i*h_2(k)) \% 7$

Running Time:
Linear Probing:
- Successful: **$\frac{1}{2}(1 + 1/(1-\alpha))$**
- Unsuccessful: **$\frac{1}{2}(1 + 1/(1-\alpha))^2$**

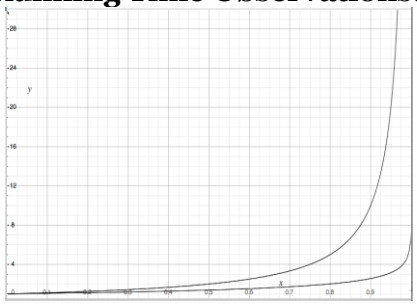
Double Hashing:
- Successful: **$1/\alpha * \ln(1/(1-\alpha))$**
- Unsuccessful: **$1/(1-\alpha)$**

Separate Chaining:
- Successful: **$1 + \alpha/2$**
- Unsuccessful: **$1 + \alpha$**

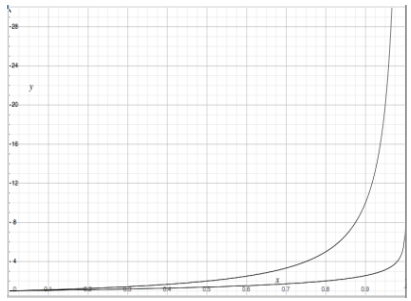
Running Time Observations:
1. As **α** increases:

2. If **α** is held constant:

Running Time Observations:



Linear Probing:
Successful: **$\frac{1}{2}(1 + 1/(1-\alpha))$**
Unsuccessful: **$\frac{1}{2}(1 + 1/(1-\alpha))^2$**



Double Hashing:
Successful: **$1/\alpha * \ln(1/(1-\alpha))$**
Unsuccessful: **$1/(1-\alpha)$**

ReHashing:
What happens when the array fills?

Better question and algorithm:

Which collision resolution strategy is better?
- Big Records:

- Structure Speed:

What structure do hash tables replace?

What constraint exists on hashing that doesn't exist with BSTs?

Why talk about BSTs at all?

CS 225 – Things To Be Doing:
1. Exam #8 (programming, MP4-like and AVL) starts Monday **2.** MP5 is available now; extra credit +7 deadline is Monday, Oct. 30 **3.** lab_btree due on Sunday (Oct. 29) **4.** Daily POTDs