

Chapter 6, Hash tables:

- ▶ General introduction; separate chaining (last week Monday)
- ▶ Open addressing (last week Wednesday)
- ▶ Hash functions (last week Friday)
- ▶ Perfect hashing (Monday)
- ▶ Hash table performance (**Today**)

Today:

- ▶ Elements of hashtable performance
- ▶ Separate chaining performance
- ▶ Open addressing performance

	Find	Insert	Delete
Unsorted array	$\Theta(n)$	$\Theta(1)$ [$\Theta(n)$]	$\Theta(n)$
Sorted array	$\Theta(\lg n)$	$\Theta(n)$	$\Theta(n)$
Linked list	$\Theta(n)$	$\Theta(1)$	$\Theta(1)$
Balanced BST	$\Theta(\lg n)$	$\Theta(1)$ [$\Theta(\lg n)$]	$\Theta(1)$ [$\Theta(\lg n)$]
What we want	$\Theta(1)$	$\Theta(1)$	$\Theta(1)$

Find	Search the data structure for a given key
Insert	Add a new key to the data structure
Delete	Get rid of a key and fix up the data structure

`containsKey()` Find

`get()` Find

`put()` Find + insert

`remove()` Find + delete

$$\begin{array}{r}
 O(1) \quad c_0 \\
 O(1) \quad c_0 \\
 O(1) \quad c_0 \\
 \vdots \\
 O(1) \quad c_0 \\
 \text{rehash} \longrightarrow O(n) \quad c_1 + c_2 n \\
 O(1) \quad c_0 \\
 \vdots \\
 O(1) \quad c_0
 \end{array}
 \left. \vphantom{\begin{array}{r}
 O(1) \quad c_0 \\
 O(1) \quad c_0 \\
 O(1) \quad c_0 \\
 \vdots \\
 O(1) \quad c_0 \\
 O(n) \quad c_1 + c_2 n \\
 O(1) \quad c_0 \\
 \vdots \\
 O(1) \quad c_0
 \end{array}} \right\}
 \begin{array}{l}
 T(n) = (n-1)c_0 + c_1 + c_2 n \\
 = (c_0 + c_2)n + c_1 - c_0 \\
 = \Theta(n)
 \end{array}$$





$$\frac{(n+1) + n + (n-1) + \cdots + 3 + 2 + \overbrace{1 + \cdots + 1}^{m-n}}{m}$$

$$= \frac{m + n + (n-1) + \cdots + 2 + 1}{m} \quad \text{the initial } m \text{ accounting for the last probe in each case}$$

$$= \frac{m}{m} + \frac{(n+1) \cdot \frac{n}{2}}{m} \quad \text{as an arithmetic series}$$

$$\approx 1 + \frac{(n+1) \cdot \frac{n}{2}}{2 \cdot n} \quad \text{since } m \text{ is about } 2 \cdot n$$

$$= 1 + \frac{n+1}{4} \quad \text{by cancellation}$$



$$\frac{[(s_0 + 1) + s_0 + (s_0 - 1) + \cdots + 2] + \cdots + 1 + \cdots + 1}{m} = 1 + \frac{\sum_{i=0}^{\gamma-1} \sum_{j=1}^{s_i} j}{m}$$

What is the probability that a miss k requires at least i probes?



Conditional probability

$P(X | Y)$: What is the probability of event X in light of event Y ?

$$P(X \wedge Y) = P(X) \cdot P(X | Y)$$

$$P(X_0 \wedge X_1 \wedge \dots \wedge X_{N-1}) = P(X_0) \cdot P(X_1 | X_0) \cdot P(X_2 | X_0 \wedge X_1) \cdot \dots \cdot P(X_{N-1} | X_0 \wedge \dots \wedge X_{N-2})$$



$$P(T[h(k) + 1] \neq \text{null} \mid T[h(k)] \neq \text{null}) = \frac{n - 1}{m - 1}$$

The probability that a miss requires at least i probes:

$$\begin{aligned} \frac{n}{m} \cdot \frac{n - 1}{m - 1} \cdots \frac{n - i + 2}{m - i + 2} \\ \leq \left(\frac{n}{m}\right)^{i-1} & \text{ since } n < m \\ \leq \alpha^{i-1} & \text{ by substitution} \end{aligned}$$

$$\sum_{i=1}^m i \cdot P\left(\begin{array}{c} \text{it takes} \\ i \text{ probes} \end{array}\right) = \sum_{i=1}^m i \cdot \left(P\left(\begin{array}{c} \text{it takes} \\ \text{at least } i \\ \text{probes} \end{array}\right) - P\left(\begin{array}{c} \text{it takes at} \\ \text{least } i+1 \\ \text{probes} \end{array}\right) \right)$$

$$= \sum_{i=1}^m P\left(\begin{array}{c} \text{it takes} \\ \text{at least } i \\ \text{probes} \end{array}\right)$$

by telescoping

$$\leq \sum_{i=1}^m \alpha^{i-1}$$

by the previous result

$$\leq \sum_{i=1}^{\infty} \alpha^{i-1}$$

since $m < \infty$

$$= \sum_{i=0}^{\infty} \alpha^i$$

by a change of variable

$$= \frac{1}{1 - \alpha}$$

by geometric series

Is the following assumption true for linear probing?

$$P(T[h(k) + 1] \neq \text{null} \mid T[h(k)] \neq \text{null}) = \frac{n - 1}{m - 1}$$

In general, is the following assumption true for a probing strategy?

$$P(T[\sigma(k, 1)] \neq \text{null} \mid T[\sigma(k, 0)] \neq \text{null}) = \frac{n - 1}{m - 1}$$

What is the difference between

Each array index is
equally likely to be
the hash of a given key.

vs

Each array position is
equally likely to be
occupied.

Linear probing is biased towards clustering:



x	Number of buckets with exactly x previous buckets filled	Number of filled buckets with exactly x previous buckets filled	Probability that a bucket is filled if exactly x previous buckets are filled.
0	97	48	.495
1	48	22	.458
2	22	12	.545
3	12	7	.583
4	7	4	.571
5	4	3	.75
6	3	2	.667
7	2	2	1
8	2	0	0

Expected number of probes for a miss in a hashtable using linear probing (from Knuth):

$$\frac{1}{2} \cdot \left(1 + \frac{1}{(1 - \alpha)^2} \right)$$

After n calls to `put()` with unique keys, no removals, consider **average chain length** over all keys (low is good), **percent of keys that are in their ideal location** (high is good), and **length of the longest chain** (low is good)

	n	Linear probing			Quadratic probing			Double hashing		
Surnames	1000	2.092	64.7%	31	1.421	75.8%	9	2.327	65.2%	31
Mountains	1360	1.568	73.8%	17	1.729	65.8%	11	1.770	73.4%	16
Mountains (height)	1360	1.932	75.1%	99	1.882	68.9%	18	1.830	72.4%	13
Chemicals	663	1.517	75.0%	16	1.729	65.5%	10	1.701	75.5%	9
Chemicals (symbol)	663	1.885	71.0%	20	1.837	66.4%	13	1.798	72.7%	12
Books	718	1.419	76.7%	8	1.659	70.0%	11	1.656	75.8%	8
Books (ISBN)	718	1.542	74.4%	21	1.670	67.8%	15	1.724	74.5%	10
Random strings	5000	1.544	77.6%	49	1.735	69.9%	37	1.598	78.1%	13
Random strings	5000	1.531	77.1%	35	1.729	69.8%	28	1.593	77.9%	12
Random strings	5000	1.643	77.5%	76	1.754	68.6%	29	1.590	78.1%	13

Coming up:

Do **Open Addressing Hashtable** project (suggested by **today** Apr 19)

Do **Perfect Hashing** project (suggested by Wednesday, Apr 26)

Due Today, Wed, Apr 19 (end of day)

Review end of Section 7.3 (not Section 7.6, which doesn't exist)

Take quiz

Due Fri, Apr 21 (end of day)

Read Section 8.1

Do Exercises 8.(4 & 5)

Take quiz

(Accompanying programming practice problem and quiz due next week Monday)

Due Mon, Apr 25 (end of day)

Do exercises 8.(7, 14, & 20) (from Section 8.1)

Read Section 8.2