Chapter 7, Hash tables:

- General introduction; separate chaining (Friday, Nov 18)
- Open addressing (Monday before Thanksgiving)
- Hash table performance (**Today**)
- (Begin Chapter 8, Strings (Wednesday))

Today:

- Elements of hashtable performance
- Clustering and chaining in open addressing
- The mathematics of hash functions
- Perfect hashing

Coming up:
Do **Open Addressing** *project (suggested by Friday, Dec2)*

Due **Today, Nov 28** *(end of day) (recommended to have been done before break)*
Read Section 7.3
Do Exercises 7.(4,5,7,8)
Take quiz (on Section 7.3 etc)

Due **Wed, Nov 30** *(end of day)*
Read Section 8.1
Do Exercises 8.(4 & 5)

Due **Thurs, Dec 1**
Take quiz (on Section 8.1)

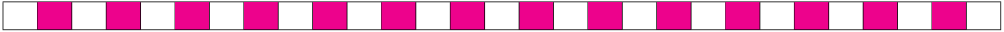
Due **Fri, Dec 2**
Do Exercises 8.(7, 14, 20)
Read Section 8.2

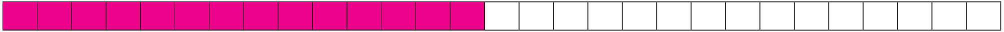
Find	Search the data structure for a given key
Insert	Add a new key to the data structure
Delete	Get rid of a key and fix up the data structure

`containsKey()`	Find
`get()`	Find
`put()`	Find + insert
`remove()`	Find + delete

	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	$\Theta(1)$	$\Theta(1)$	$\Theta(1)$

$$
\left.\begin{array}{rll}
 & O(1) & c_0 \\
 & O(1) & c_0 \\
 & O(1) & c_0 \\
 & \vdots & \\
 & O(1) & c_0 \\
\text{rehash} \longrightarrow & O(n) & c_1 + c_2 n \\
 & O(1) & c_0 \\
 & \vdots & \\
 & O(1) & c_0
\end{array}\right\}
\begin{array}{rcl}
T(n) & = & (n-1)c_0 + c_1 + c_2 n \\
 & = & (c_0 + c_2)n + c_1 - c_0 \\
 & = & \Theta(n)
\end{array}
$$





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

$$\begin{aligned}
&= \frac{m+n+(n-1)+\cdots+2+1}{m} && \text{the initial } m \text{ accounting for the last probe in each case}\\
&= \frac{m}{m}+\frac{(n+1)\cdot\frac{n}{2}}{m} && \text{as an arithmetic series}\\
&\approx 1+\frac{(n+1)\cdot\frac{n}{2}}{2\cdot n} && \text{since } m \text{ is about } 2\cdot n\\
&= 1+\frac{n+1}{4} && \text{by cancellation}
\end{aligned}$$

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

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

Conditional probability

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

$$
\begin{aligned}
P(X \wedge Y) &= P(X) \cdot P(X \mid Y) \\
P(X_0 \wedge X_1 \wedge \cdots \wedge X_{N-1}) &= P(X_0) \cdot P(X_1 \mid X_0) \cdot P(X_1 \mid X_0 \wedge X_1) \cdots P(X_{N-1} \mid X_0 \wedge \cdots \wedge X_{N-2})
\end{aligned}
$$

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

The probability that a miss requires at least i probes:

$$\frac{n}{m} \cdot \frac{n-1}{m-1} \cdots \frac{n-i+2}{m-i+2}$$

$$\leq \quad \left(\frac{n}{m}\right)^{i-1} \qquad \text{since } n < m$$

$$\leq \quad \alpha^{i-1} \qquad \text{by substitution}$$

$$
\begin{aligned}
\sum_{i=1}^{m} i \cdot P\begin{pmatrix}\text{it takes}\\ i \text{ probes}\end{pmatrix} &= \sum_{i=1}^{m} i \cdot \left(P\begin{pmatrix}\text{it takes}\\ \text{at least } i\\ \text{probes}\end{pmatrix} - P\begin{pmatrix}\text{it takes at}\\ \text{least } i+1\\ \text{probes}\end{pmatrix}\right) \\
&= \sum_{i=1}^{m} P\begin{pmatrix}\text{it takes}\\ \text{at least } i\\ \text{probes}\end{pmatrix} && \text{by telescoping} \\
&\le \sum_{i=1}^{m} \alpha^{i-1} && \text{by the previous result} \\
&\le \sum_{i=1}^{\infty} \alpha^{i-1} && \text{since } m < \infty \\
&= \sum_{i=0}^{\infty} \alpha^{i} && \text{by a change of variable} \\
&= \frac{1}{1-\alpha} && \text{by geometric series}
\end{aligned}
$$

Is the following assumption true for linear probing?

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

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

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

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

Hash functions should distribute the keys *uniformly* and *independently*.

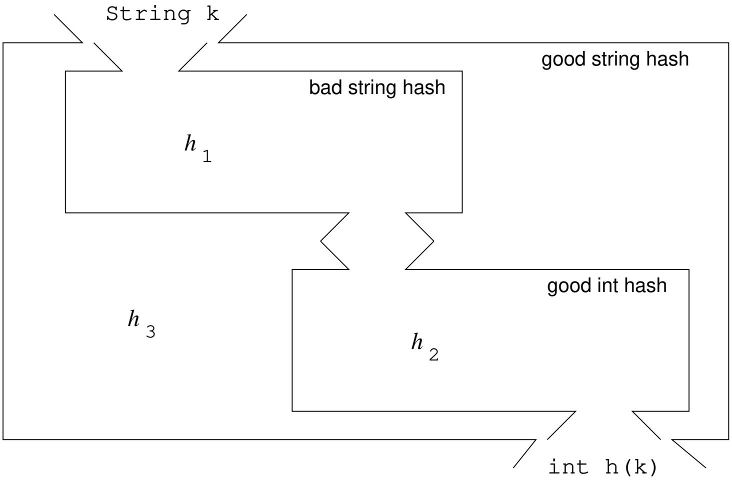
Uniformity:

$$P(h(k) = i) = \frac{1}{m}$$

Independence:

$$P(h(k_1) = i) \quad = \quad P(h(k_1) = i \mid h(k_2) = j)$$

Why do we talk about integer hashes?



Division method:

$$h(k) = k \mod m$$

Middle square method (see code)

Multiplicative method:

$$h(k) = \lfloor m(k \cdot a - \lfloor k \cdot a \rfloor) \rfloor$$

“Universal” hash (later...)

ASCII sum:

$$h(k) = \left(\sum_{i=0}^{n-1} s[i] \right)$$

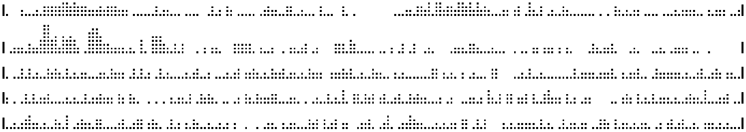
String polynomial:

$$h(k) = (k[0] \cdot b^{n-1} + k[1] \cdot b^{n-2} + \cdots + k[n-2] \cdot b + k[n-1]) \mod m$$

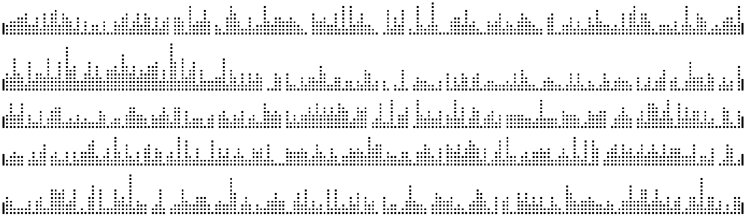
Carter-Wegman:

$$\begin{aligned} h(k) &= (h_0(k[0]) + h_1(k[1]) + \cdots + h_{n-1}(k[n-1])) \mod m \\ &= \left(\sum_{i=0}^{n-1} h_i(k[i]) \right) \mod m \end{aligned}$$

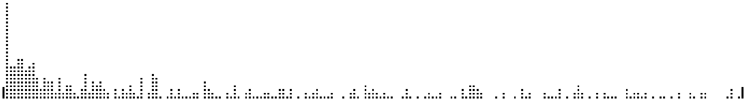
Area codes ($n = 303$)	Average penalty	Variance
Division	.673	.808
Mid square	1.09	1.64
Multiplicative	.508	.478
Fibonacci	.617	.696
Universal	.578	.617

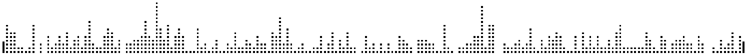


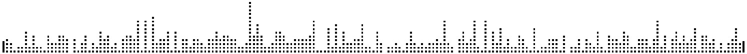
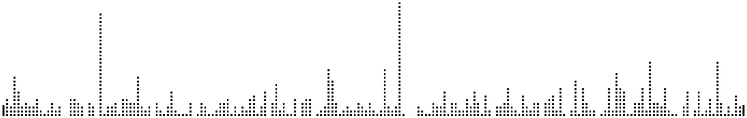
Book ISBNs ($n = 718$)	Average penalty	Variance
Division	.618	1.05
Mid square	.812	1.48
Multiplicative	.565	.954
Fibonacci	.544	.873
Universal	.667	1.15



Randomly generated from $[0, 1000)$ $(n = 150)$		Average penalty	Variance
Division		1.36	.958
Mid square		1.86	1.96
Multiplicative		1.34	.919
Fibonacci		1.41	1.07
Universal		1.39	1.02

Randomly generated from $[0, 1000)$ $(n = 400)$			
Division		.518	1.16
Mid square		1.73	3.68
Multiplicative		.405	.930
Fibonacci		.448	.980
Universal		.488	1.08

Chemicals ($n = 663$)		Average penalty	Variance
ASCII sum		.505	1.00
String polynomial		.424	.805
Carter-Wegman		.800	1.63

Books ($n = 718$)			
ASCII sum		.818	1.51
String polynomial		.745	1.30
Carter-Wegman		2.06	4.08

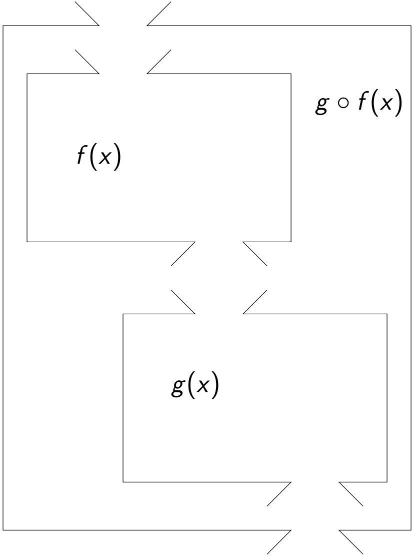
Randomly generated strings ($n = 150$)		Average penalty	Variance
ASCII sum		1.32	.879
String polynomial		1.43	1.09
Carter-Wegman		1.41	1.05

Randomly generated strings ($n = 400$)			
ASCII sum		.515	1.15
String polynomial		.425	.925
Carter-Wegman		.540	1.20

A hashing scheme must reduce the occurrence of collisions and "deal" with them when they happen.

- *Separate chaining*, where $m < n$, deals with collisions by chaining keys together in a bucket.
- *Open addressing*, where $n < m$, deals with collisions by finding an alternate location.
- *Perfect hashing* deals with collisions by preventing them altogether.

This topic is parallel with the *optimal BST problem*: What if we knew the keys ahead of time? What if we got to choose the hash function based on what keys we have?



Let $\mathscr{H}$ stand for a *class* of hash functions (a set of hash functions defined by some formula).

Let m be the number of buckets.

$\mathscr{H}$ is *universal* if

$$\forall\, k, \ell \in Keys,\ \ |\{h \in \mathscr{H} \mid h(k) = h(\ell)\}| \leq \frac{|\mathscr{H}|}{m}$$

$\mathscr{H}$ is *universal* if

$$\forall\ k, \ell \in Keys, \quad |\{h \in \mathscr{H} \mid h(k) = h(\ell)\}| \leq \frac{|\mathscr{H}|}{m}$$

One particular *family* of *classes* of hash functions, given p, a prime number greater than all keys, and m, the number of buckets, is denoted $\mathscr{H}_{mp}$:

$$\mathscr{H}_{mp} = \{\ h_{ab}(k) = ((ak + b) \mod p) \mod m \mid a \in [1, p) \text{ and } b \in [0, p)\}$$

Theorem $\mathscr{H}_{pm}$ *is universal.*

Proof. *Suppose* p *and* m *as specified earlier. Suppose* $k, \ell \in$ *Keys, and* $h_{ab} \in \mathscr{H}_{pm}$ *(which implies supposing that* $a \in [1, p)$ *and* $b \in [0, p)$*).*
Let $r = (a \cdot k + b) \mod p$ *and* $s = (a \cdot \ell + b) \mod p$
Subtracting gives us

$$\begin{aligned} r - s &\equiv (a \cdot k + b) - (a \cdot \ell + b) && \mod p \\ &\equiv a \cdot (k - \ell) && \mod p \end{aligned}$$

Now a *cannot be 0 because* $a \in [1, p)$*. Similarly* $k - \ell$ *cannot be 0, since* $k \neq \ell$*. Hence* $a \cdot (k - \ell) \neq 0$*.*
Since p *is prime and greater than* a*,* k*, and* ℓ*, it cannot be a factor of* $a \cdot (k - \ell)$*.*
In other words, $a \cdot (k - \ell) \mod p \neq 0$*. By substitution,* $r - s \neq 0$*, and so* $r \neq s$*.*
By another substitution, $(a \cdot k + b) \mod p \neq (a \cdot \ell + b) \mod p$*.*

Define the following function, given k and ℓ, which maps from (a, b) pairs to (r, s) pairs (formally, $[1, p) \times [0, p) \to [1, p) \times [0, p)$):

$$\phi_{k\ell}(a, b) = ((a \cdot k + b) \mod p, (a \cdot \ell + b) \mod p)$$

Now consider the inverse of that function.

$$\begin{aligned} \phi_{k\ell}^{-1}(r, s) &= (((r - s) \cdot (k - \ell)^{-1}) \mod p), (r - ak) \mod p) \\ &= (a, b) \end{aligned}$$

The existence of ϕ^{-1} implies that ϕ is a one-to-one correspondence. Hence for each (a, b) pair, there is a unique (r, s) pair. Since the pair (a, b) specifies a hash function, that means that for each hash function in the family $\mathscr{H}_{pm}$, there is a unique (r, s) pair.

There are $p-1$ possible choices for a and p choices for b, so there are $p\cdot(p-1)$ hash functions in family $\mathscr{H}_{pm}$. Likewise there are p choices for r, and for each r there are $p-1$ choices for s (since $s \neq r$). Thus we can partition the set $\mathscr{H}_{pm}$ into p subsets by r value, each subset having $p-1$ hash functions.
For a given r, at most one out of every m can have an s that is equivalent to r mod m, in other words, at most $\frac{p-1}{m}$ hash functions.
Now sum that for all p of the subsets of $\mathscr{H}_{pm}$, and we find that the number of hash functions for which k and ℓ collide are

$$p \cdot \frac{p-1}{m} = \frac{p \cdot (p-1)}{m} = \frac{|\mathscr{H}_{pm}|}{m}$$

Therefore $\mathscr{H}_{pm}$ is universal by definition. $\square$

Theorem [Probability of any collisions.] *If Keys is a set of keys, $m = |Keys|^2$, p is a prime greater than all keys, and $h \in \mathscr{H}_{pm}$, then the probability that any two distinct keys collide in h is less than $\frac{1}{2}$.*

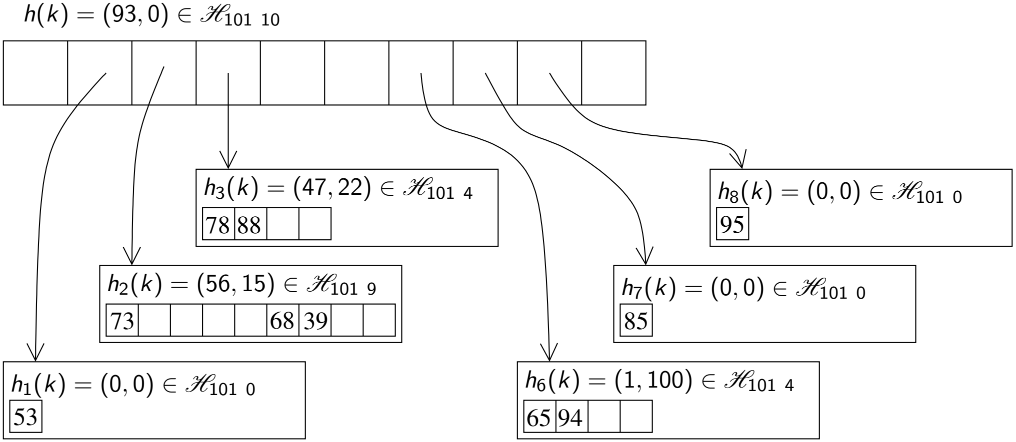
Proof. *Suppose we have a set Keys, $m = |Keys|^2$, p is a prime greater than all keys, and $h \in \mathscr{H}_{pm}$.*
Consider the number of pairs of unique keys. The number of pairs of keys is

$$\binom{n}{2} = \frac{n!}{2! \cdot (n-2)!} = \frac{n!}{2 \cdot (n-2)!} = \frac{n \cdot (n-1) \cdot \cancel{(n-2)!}}{2 \cdot \cancel{(n-2)!}} = \frac{n \cdot (n-1)}{2}$$

Since $\mathscr{H}_{pm}$ is universal, each pair collides with probability $\frac{1}{m}$. Multiply that by the number of pairs, and the expected number of collisions is

$$\begin{aligned}
\frac{n\cdot(n-1)}{2}\cdot\frac{1}{m} &< \frac{n^2}{2}\cdot\frac{1}{m} && \textit{since } n\cdot(n-1) < n^2 \\
&= \frac{n^2}{2}\cdot\frac{1}{n^2} && \textit{since } m = n^2 \\
&= \frac{1}{2} && \textit{by cancelling } n^2
\end{aligned}$$

With the expected number of collisions less than one half, the probability there are any collisions is also less than $\frac{1}{2}$. $\square$



$h(k) = (93, 0) \in \mathscr{H}_{101\ 10}$
$h_3(k) = (47, 22) \in \mathscr{H}_{101\ 4}$
78 88
$h_8(k) = (0, 0) \in \mathscr{H}_{101\ 0}$
95
$h_2(k) = (56, 15) \in \mathscr{H}_{101\ 9}$
73 68 39
$h_7(k) = (0, 0) \in \mathscr{H}_{101\ 0}$
85
$h_1(k) = (0, 0) \in \mathscr{H}_{101\ 0}$
53
$h_6(k) = (1, 100) \in \mathscr{H}_{101\ 4}$
65 94

Coming up:

Do **Open Addressing** *project (suggested by Friday, Dec2)*

Due **Today, Nov 28** *(end of day) (recommended to have been done before break)*
Read Section 7.3
Do Exercises 7.(4,5,7,8)
Take quiz (on Section 7.3 etc)

Due **Wed, Nov 30** *(end of day)*
Read Section 8.1
Do Exercises 8.(4 & 5)

Due **Thurs, Dec 1**
Take quiz (on Section 8.1)

Due **Fri, Dec 2**
Do Exercises 8.(7, 14, 20)
Read Section 8.2