Chapter 6, Hash tables:

- General introduction; separate chaining (last week Friday)
- Open addressing (Wednesday)
- Separate chaining revisited (lab Thursday)
- Hash functions (**Today**)
- Perfect hashing (next week Monday)
- Hash table performance (next week Wednesday)

Today:

- Lab retrospective
- Hash function properties
- Integer hashes
- String hashes
- Experimental results

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

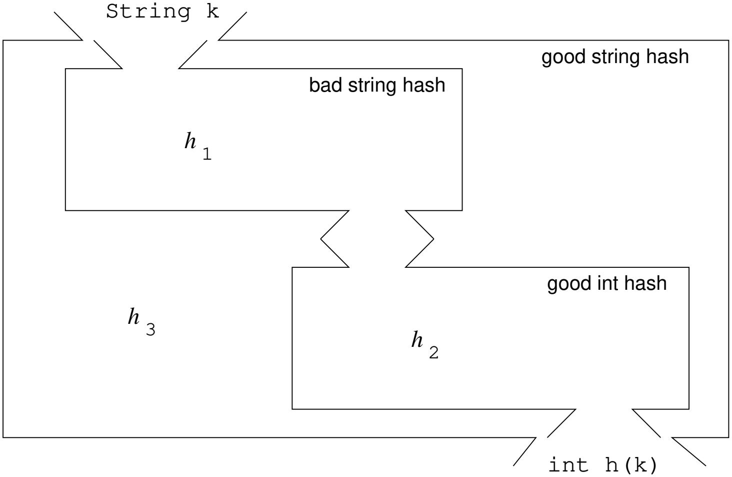
Uniformity:

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

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:

	Decimal	Binary
Original	37,914	0000 0000 0000 0000 1001 0100 0001 1010
Squared	1,437,471,396	0101 0101 1010 $\underbrace{\text{1110 0001}}_{\text{middle bits}}$ 0010 1010 0100
Middle 10 bits	225	0000 0000 0000 0000 0000 0000 1110 0001

Multiplicative method:

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

“Universal” hash (next time)

ASCII sum:

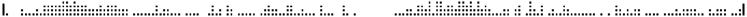
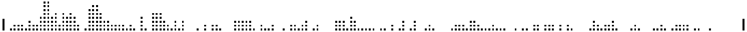
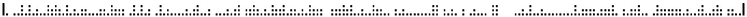
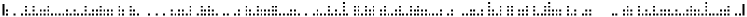
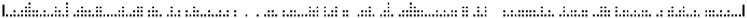
$$h(k) = \left(\sum_{i=0}^{n-1} s[i] \right)$$

String polynomial:

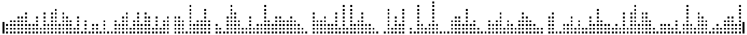
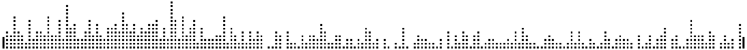
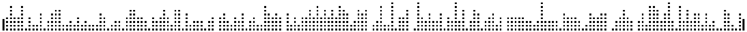
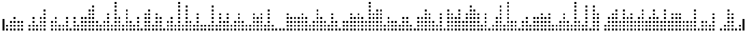
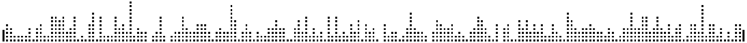
$$h(k) = (k[0] \cdot b^{n-1} + k[1] \cdot b^{n-2} + \cdots + k[n-2] \cdot b + k[n-1]) \mod m$$

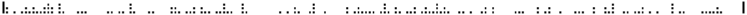
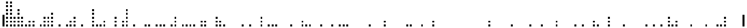
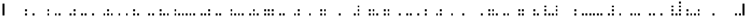
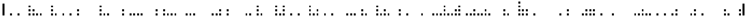
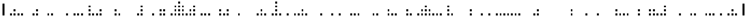
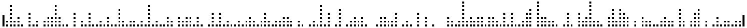
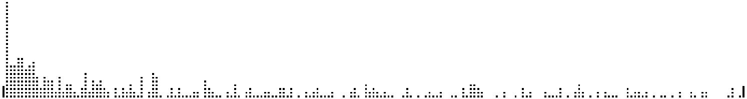
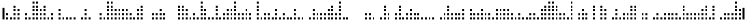
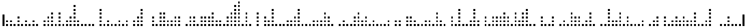
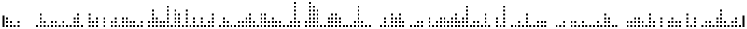
Carter-Wegman:

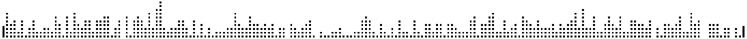
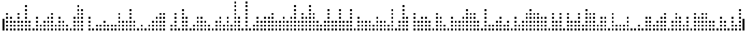
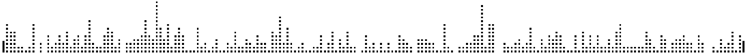
$$\begin{aligned} h(k) &= (h_0(k[0]) + h_1(k[1]) + \cdots + h_{n-1}(k[n-1])) \mod m \\ &= \left(\textstyle\sum_{i=0}^{n-1} h_i(k[i]) \right) \mod m \end{aligned}$$

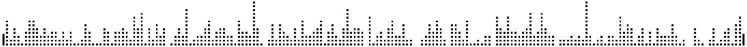
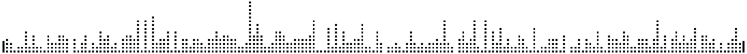
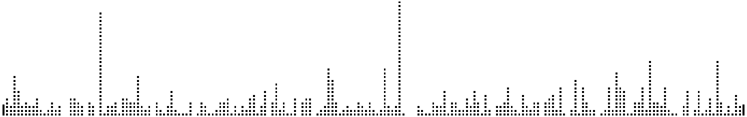
Area codes ($n = 303$)		Average penalty	Variance
Division		.673	.808
Mid square		1.09	1.64
Multiplicative		.508	.478
Fibonacci		.617	.696
Universal		.578	.617

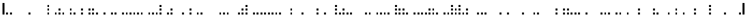
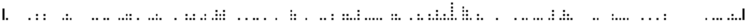
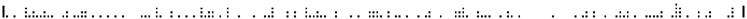
Book ISBNs ($n = 718$)

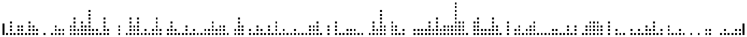
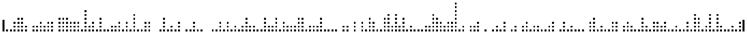
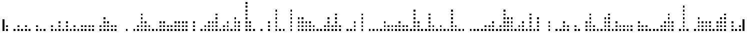
Division		.618	1.05
Mid square		.812	1.48
Multiplicative		.565	.954
Fibonacci		.544	.873
Universal		.667	1.15

		Average penalty	Variance
Randomly generated from $[0, 1000)$ $(n = 150)$			
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

Coming up:

Do **Open addressing with linear probing** *project (due Monday, Apr 15)*

Due **Mon, Apr 15**
Read Sections 7.(4 & 5)
(No practice problems or quiz)