Hash functions for strings

- transform characters to numbers, e.g. ASCII codes

  E g g e r

  69 103 103 101 114

- compute hash value from string, e.g. Java String hash for string s

  $$h = s[0] * 31^{n-1} + s[1] * 31^{n-2} + \dots + s[n-1]$$

Code

```
h := 0
for i := 0 to n-1 do
    h := 31*h + s.charAt(i)
```

E g g e r

$$69 * 31^4 + 103 * 31^3 + 103 * 31^2 + 101 * 31 + 114$$

Properties:

- $h('pot') \neq h('top')$
-

Collision Resolution by Open Addressing

- No lists
- All entries stored in table $\Rightarrow n \le m$
- Insert: probe table until empty slot is found
- Modify h:

$$h : \underset{\substack{\text{universe} \\ \text{of keys}}}{U} \times \underset{\text{probe \#}}{\{0, 1, \ldots, m-1\}} \longrightarrow \underset{\text{slot}}{\{0, \ldots, m-1\}}$$

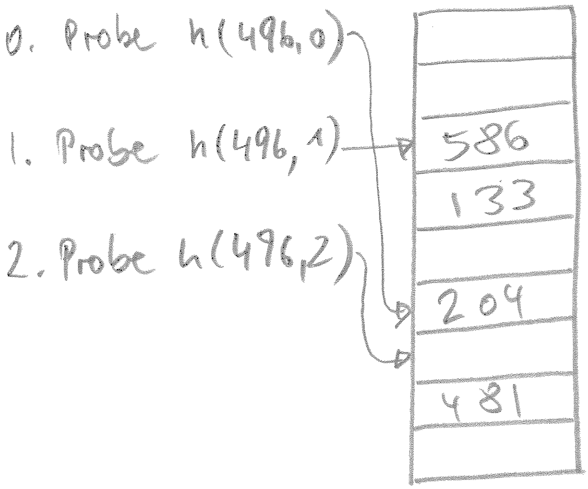
- Probe sequence for k

  $h(k,0), \quad h(k,1), \ \ldots, \ h(k, m-1)$

  should be permutation of $\{0, \ldots, m-1\}$

- Deletion: what if we create a gap?
- Full table: what if $n \ge m$?

Example : Insert $k = 496$



Search (496)

- use the same probe sequence
- successful, if entry is found
- fail, if nil is found

Optimization: remember longest probe sequence for insertion

Probing Strategies

- Linear Probing

$$h(k,i) := (h(k,0) + i) \bmod m$$

$$= (h'(k) + i) \bmod m$$

simple hashing function

Problem: Primary Clustering

Example: $m = 13$, $h(k) = k \bmod 13$

Insert: $k =$	18	41	22	44	59	32	31	73
$h(k) =$	5	2	9	5	7	6	5	8
#P:	1	1	1	2	1	2	6	3

tot = 17

0	
1	
2	41
3	
4	
5	18
6	44
7	59
8	32
9	22
10	31
11	73
12	

- Quadratic Probing

$$h(k,i) := (h'(k) + c_1 i + c_2 i^2) \bmod m$$

- Double Hashing

$$h(k,i) := (\underbrace{h_1(k)}_{\text{1st probe}} + i\, \underbrace{h_2(k)}_{\text{offset}}) \bmod m$$

Ensure permutation property: $\gcd(m, h_2(k)) = 1$

Ex: $m = 2^r$, $h_2(k)$ odd

m prime, $h_2(k) \in \{1, \dots, m-1\}$

<u>Example</u>: $h_1(k) = k \bmod 13$
$h_2(k) = 8 - (k \bmod 8)$

Insert:

$k=$	18	41	22	44	59	32	31	73
$h_1(k)=$	5	2	9	5	7	6	5	8
$h_2(k)=$	6	7	2	4	5	8	1	7
# probes	1	1	1	3	1	1	6	4

tot = 18

0	44
1	
2	41
3	73
4	
5	18
6	32
7	59
8	31
9	22
10	
11	
12	

Deletion:

- Mark deleted slot as "deleted"
- Increased retrieval time due to "jumps" on deleted slots
- Ops do no more depend on load factor

Analysis of Open Addressing

Uniform hashing assumption:

Each key is uniformly likely to have any of the $m!$ permutations as its probe sequence, independent of other keys

Unsuccessful Search

- 1 probe always needed
- with probability n/m collision $\Rightarrow$ 2nd probe
  — " — $n-1/m-1$ — " — $\Rightarrow$ 3rd probe
  ... ...
- Note $\frac{n-i}{m-i} < \frac{n}{m} = \alpha$ f.a. $i = 1, 2, \dots, n-1$

Expected value of # probes

$$1 + \frac{n}{m}\left(1 + \frac{n-1}{m-1}\left(1 + \frac{n-2}{m-2}\left(\cdots\left(1 + \frac{1}{m-n+1}\right)\cdots\right)\right)\right)$$

$$\leq 1 + \alpha\left(1 + \alpha\left(1 + \alpha\left(\cdots\left(1 + \alpha\right)\right)\cdots\right)\right)$$

$$\leq 1 + \alpha + \alpha^2 + \alpha^3 + \cdots$$

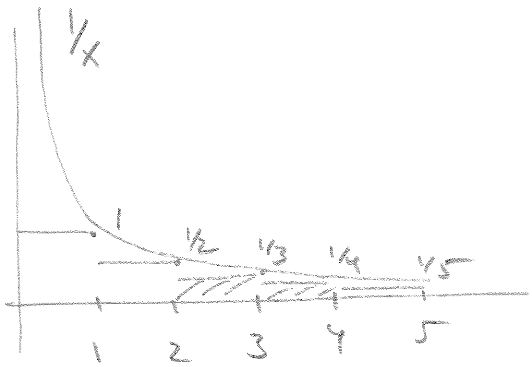
$$= \sum_{i=0}^{\infty} \alpha^i = \boxed{\frac{1}{1-\alpha}}$$

Successful Search

- Assume n keys in table
- Inserting $(i+1)$st key required $\frac{1}{1-\frac{i}{m}}$ probes (unsuccessful search befor insertion)
- Note: $\frac{1}{1-\frac{i}{m}} = \frac{m}{m-i}$
- Average for all n insertions

$$\frac{1}{n} \sum_{i=0}^{n-1} \frac{m}{m-i} = \frac{m}{n} \sum_{i=0}^{n-1} \frac{1}{m-i}$$

$$= \frac{1}{\alpha} \left(\sum_{j=m-n+1}^{m} \frac{1}{j} \right)$$



Bounding a sum by an integral

$$\le \frac{1}{\alpha} \int_{m-n}^{m} \frac{1}{x}\, dx = \frac{1}{\alpha} \left(\log m - \log(m-n) \right)$$

$$= \frac{1}{\alpha} \log \frac{m}{m-n} = \frac{1}{\alpha} \log \frac{\frac{m}{m}}{\frac{m}{m} - \frac{n}{m}} = \boxed{\frac{1}{\alpha} \log \frac{1}{1-\alpha}}$$

Analysis: Overview

	Unsucc.	Succ.
Chaining	$O(1+\alpha)$	$O(1+\alpha)$
Open Addressing	$O\left(\frac{1}{1-\alpha}\right)$	$O\left(\frac{1}{\alpha}\log\frac{1}{1-a}\right)$

O.A.

α	unsucc	succ
20 %	1.25	0.28
50 %	2	1.39
80 %	5	2.01
90 %	10	2.56