Load Factor

- Loading factor (LF), α = n / m
  n: number of keys
  m: number of slots
- If uniform distribution (1/m) to get mapped to a slot, a slot will have an expectation of α elements.
- If m increases
  - Collision decreases
  - LF decreases
  - 0.5 > LF > 0.8 is unacceptable
  - Storage requirements increases.
- Reduce collisions while keeping storage requirements low.

Linear Probing

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

- Always check the next index
- Increments index linearly with respect to i.
- Clustering problem

hash(10) = 2
hash(5) = 5
hash(15) = 7

0	72	72	72	72
1				15
2	18	18	18	18
3	43	43	43	43
4	36	36	36	36
5		10	10	10
6	6	6	6	6
7			5	5

Open Addressing – Quadratic Probing

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

- Instead of moving by one, move i^2

$c_1=0, c_2=1$
hash(89)=9
hash(18)=8
hash(49)=9
hash(49, 1) = 0
hash(58) = 8
hash(58, 1) = 9
hash(58,2) = 2
hash(69) = 9
hash(69,1) = 0
hash(69,2) = 3

0			49	49	49
1					
2				58	58
3					69
4					
5					
6					
7					
8		18	18	18	18
9	89	89	89	89	89

Example 2 – Double Hashing

	insert(76)	insert(93)	insert(40)	insert(47)	insert(10)	insert(55)
	76%7 = 6	93%7 = 2	40%7 = 5	47%7 = 5 5 - (47%5) = 3	10%7 = 3	55%7 = 6 5 - (55%5) = 5
0						
1				47	47	47
2		93	93	93	93	93
3					10	10
4						55
5			40	40	40	40
6	76	76	76	76	76	76
probes:	1	1	1	2	1	2

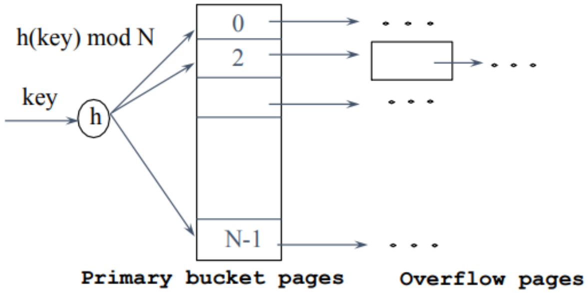
Closed Hashing >>> https://www.cs.usfca.edu/~galles/visualization/ClosedHash.html **Open Hashing >>>** https://www.cs.usfca.edu/~galles/visualization/OpenHash.html

Dynamic Hashing Methods

- *As for any index, 2 alternatives for data entries* **k***:
  - <**k**, rid of data record with search key value **k**>
  - <**k**, list of rids of data records with search key **k**>
  - Choice orthogonal to the *indexing technique*
- *Hash-based* indexes are best for *equality selections*. ***Cannot*** support range searches.

Static Hashing

- # primary pages fixed, allocated sequentially, never de-allocated; overflow pages if needed.
- **h**(*k*) mod M = bucket to which data entry with key *k* belongs. (M = # of buckets)



Extendible Hashing >>> https://www.youtube.com/watch?v=TtkN2xRAgv4&t=519s

Extendible Hashing Example