Collision Resolution by Open Addressing

Open addressing: probe array for the "next" slot which is still empty.

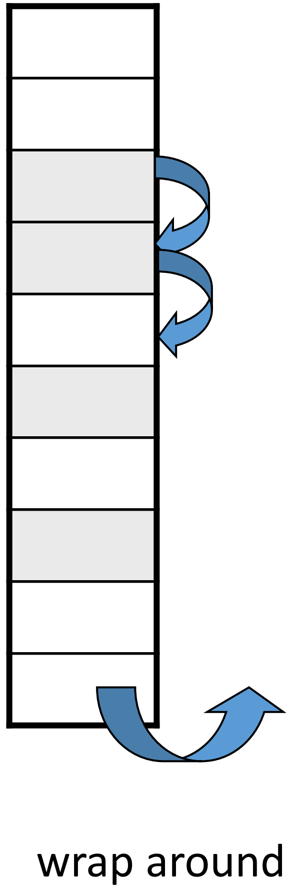
- Linear probing
- Quadratic probing
- Double hashing

Linear probing: Inserting a key

Idea: when there is a collision, check the next available position in the table (i.e., probing)

$$h(x, i) = (h_1(x) + i) \bmod k$$
$$i=0,1,2,\ldots$$

First slot probed: $h_1(x)$
Second slot probed: $h_1(x) + 1$
Third slot probed: $h_1(x)+2$, and so on



probe sequence: $\{h_1(x), h_1(x)+1, h_1(x)+2, \ldots.\}$

Linear Probing Example

	insert(14) 14%7 = 0	insert(8) 8%7 = 1	insert(21) 21%7 =0	insert(2) 2%7 = 2
0	14	14	14	14
1		8	8	8
2			21	12
3				2
4				
5				
6				

Quadratic Probing

Idea: Spread out the search for an empty slot – Increment by i^2 instead of i

$h_i(X) = (h(X) + i^2)$ % TableSize

$h_0(X) = h(X)$ % TableSize

$h_1(X) = h(X) + 1$ % TableSize

$h_2(X) = h(X) + 4$ % TableSize

$h_3(X) = h(X) + 9$ % TableSize

Quadratic Probing Example

	insert(14) 14%7 = 0	insert(8) 8%7 = 1	insert(21) 21%7 =0	insert(2) 2%7 = 2
0	14	14	14	14
1		8	8	8
2				2
3				
4			21	21
5				
6				

Double Hashing

Idea:

(1) Use one hash function to determine the first index

(2) Use a second hash function (independent of the first hash function) to the key when a collision occurs

$$h(x, i) = (h_1(x) + i \, . \, h_2(x)\ ) \bmod k, \quad i=0,1,...$$

second hash function $h_2(x) = y – (x \bmod y)$ in case of collision,

y is a prime number $< k$ (table size)

h_1 and h_2 are independent, $0 \le h_1 \le k-1$, $1 \le h_2 \le k-1$

Double Hashing

The result of the second hash function will be the number of positions form the point of collision to insert.

Requirements for the second function:

- it must never evaluate to 0
- must make sure that all cells can be probed
- it should cycle through the whole table
- it should be very fast to compute
- it should be independent of $h_1(x)$

Double Hashing Example

	insert(14) 14%7 = 0	insert(8) 8%7 = 1	insert(21) 21%7 =0 5-(21%5)=4	insert(2) 2%7 = 2	insert(7) 7%7 = 0 5-(7%5)=3
0	14	14	14	14	14
1		8	8	8	8
2				2	2
3					7
4			21	21	21
5					
6					

Double Hashing

Linear/Quadratic probing vs Double Hashing

Unlike linear probing and quadratic probing, the interval depends on the data, so that values mapping to the same location have different bucket sequences; this minimizes repeated collisions.