Chapter 6, Hash tables:

- General introduction; separate chaining (Wednesday)
- Practice open addressing (Thursday lab)
- Open addressing (**Today**)
- Hash functions (next week Monday)
- Perfect hashing (week-after Monday)
- Hash table performance (week-after Wednesday)

Today:

- Review/finish hash table concepts
- Lab retrospective
- Basic idea and example of open addressing
- Terminology, code, and invariant
- Probing strategies
- Deletion

Hash table terminology:

- Hash table: A *data structure*, not an ADT ...
- Bucket: A position in the (main) array, or, abstractly, an index in the range $[0, m)$.
- Hash function: A function from keys to buckets.
- Collision: When two keys are hashed to the same bucket.
- Chain: A sequence of keys that needs to be searched through to find a given key.
- Load factor (α): An upper bound on the ratio of keys to buckets.

Factors in best vs worst vs expected case:

- State of the table
- Length of the bucket
- Position of key in the bucket.

Parameters that can be adjusted for engineering a hash table:

- Load factor α
- Rehash strategy
- Hash function

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

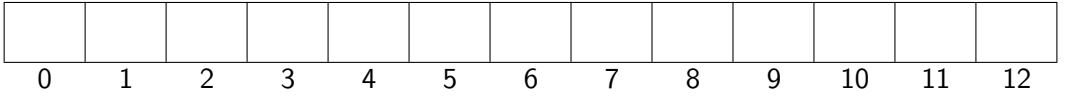
Hash functions should distribute the keys *uniformly* and *independently*.

Uniformity:

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

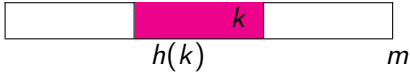
Independence:

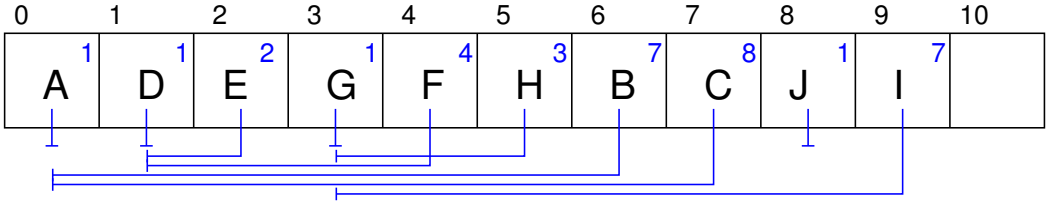
$$P(h(k_1) = i) \quad = \quad P(h(k_1) = i \mid h(k_2) = j)$$

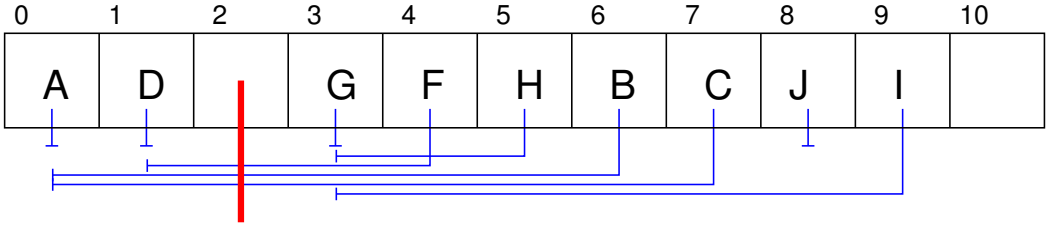

0
1
2
3
4
5
6
7
8
9
10
11
12

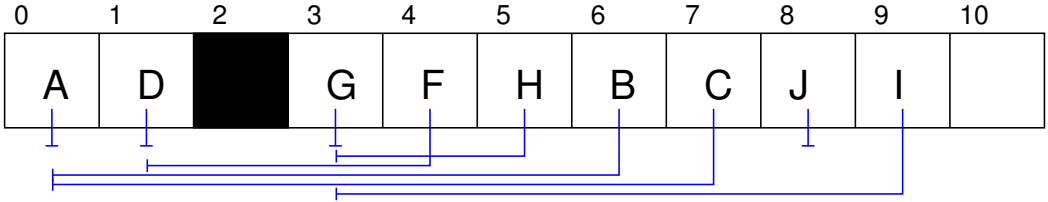
Invariant (Class `OpenAddressingHashMap`)

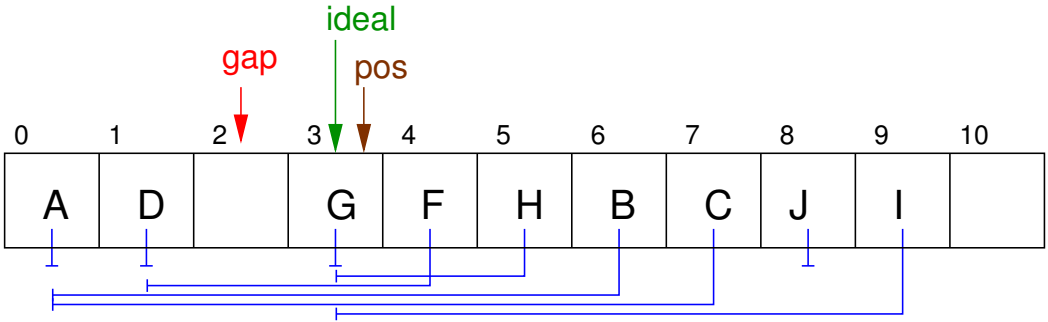
1. *The table it not full; there exists* $i \in [0, m)$ *such that* $\texttt{table}[i] = \texttt{null}$.
2. *There are no breaks in the chain for any key in the table; for all* $i \in [0, m)$ *such that* `table`$[i]$ *contains key* k,
   - *if* $h(k) \leq i$, *then for all* $j \in [h(k), i]$, $\texttt{table}[j] \neq \texttt{null}$;
   - *if* $i < h(k)$, *then for all* $j \in [0, i] \cup [h(k), m)$, $\texttt{table}[j] \neq \texttt{null}$.

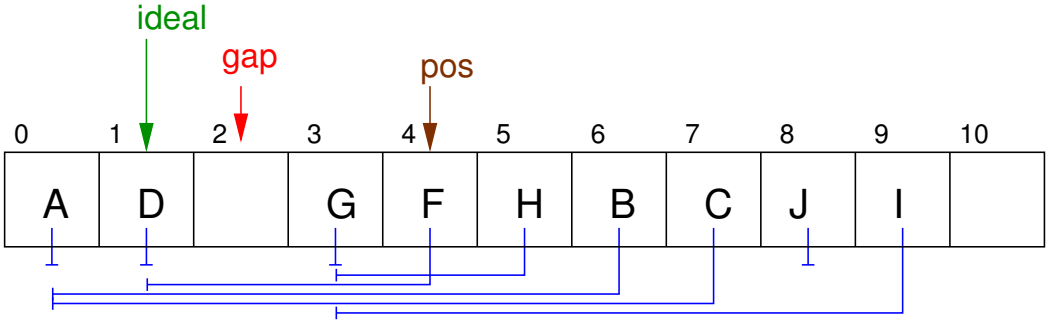


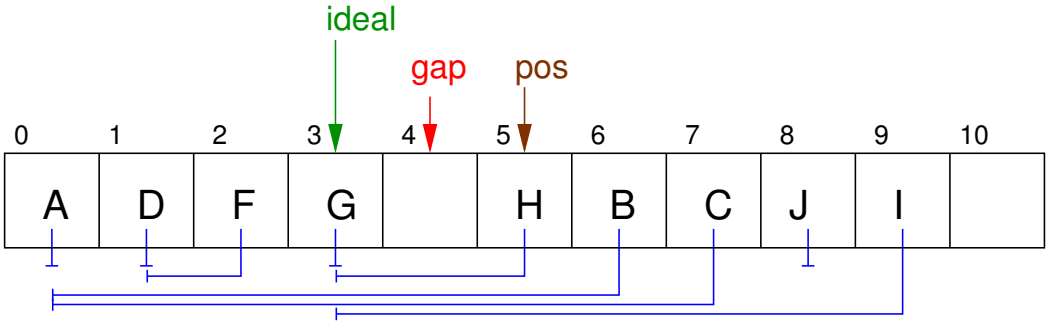

0
1
2
3
4
5
6
7
8
9
10
A
1
D
1
E
2
G
1
F
4
H
3
B
7
C
8
J
1
I
7

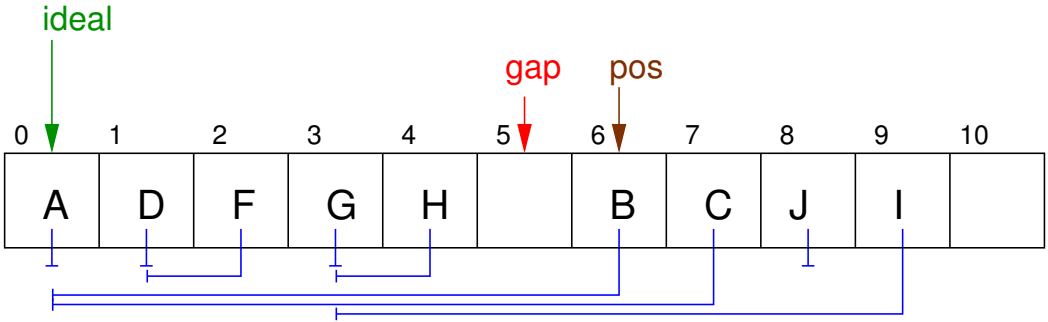

0
1
2
3
4
5
6
7
8
9
10
A
D
G
F
H
B
C
J
I

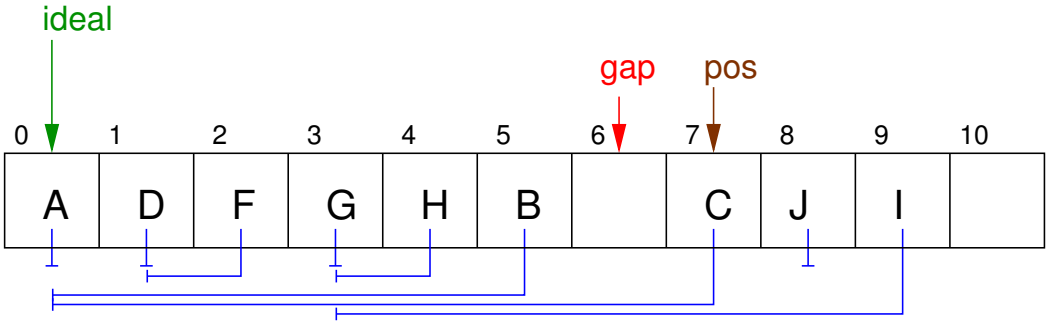

0
1
2
3
4
5
6
7
8
9
10
A
D
G
F
H
B
C
J
I

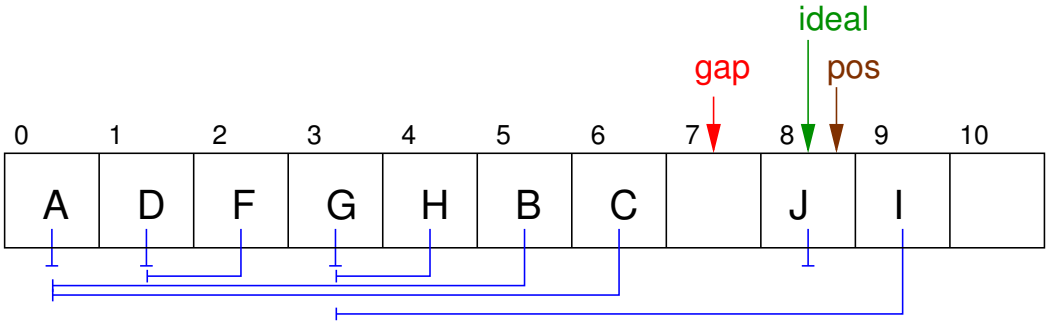

ideal
gap
pos
0
1
2
3
4
5
6
7
8
9
10
A
D
G
F
H
B
C
J
I

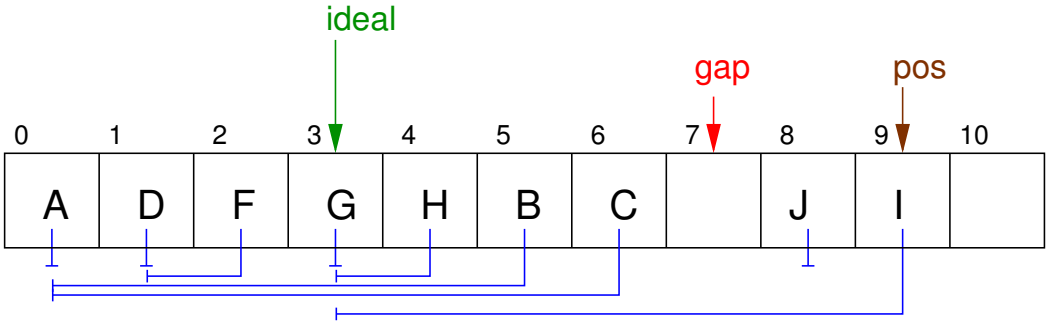

ideal
gap
pos
0
1
2
3
4
5
6
7
8
9
10
A
D
G
F
H
B
C
J
I

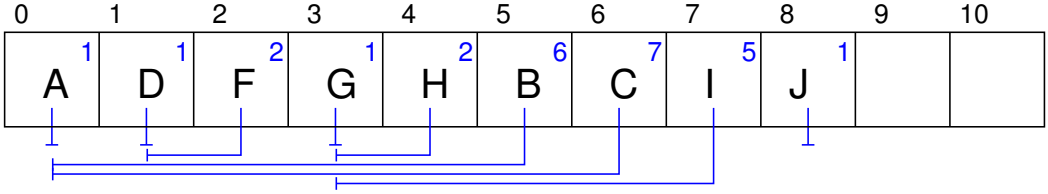

ideal
gap
pos
0
1
2
3
4
5
6
7
8
9
10
A
D
F
G
H
B
C
J
I


ideal
gap
pos
0
1
2
3
4
5
6
7
8
9
10
A
D
F
G
H
B
C
J
I

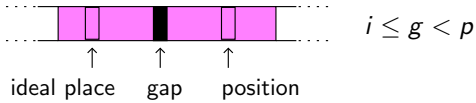

ideal
gap
pos
0
1
2
3
4
5
6
7
8
9
10
A
D
F
G
H
B
C
J
I


ideal
gap
pos
0
1
2
3
4
5
6
7
8
9
10
A
D
F
G
H
B
C
J
I

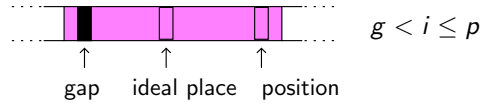

ideal
gap
pos
0
1
2
3
4
5
6
7
8
9
10
A
D
F
G
H
B
C
J
I


0
1
2
3
4
5
6
7
8
9
10
A
1
D
1
F
2
G
1
H
2
B
6
C
7
I
5
J
1

Cases to plug the gap



Cases to skip the gap



Invariant (Loop of optimized remove in linear probing.)

For all positions $k \in (i, j)$, gap is the only position, if any, between its ideal place ($h(\texttt{keys}[k])$) and its actual place (k).

Coming up:

Do **Optimal BST** *project (Due Mon, Nov 24)*
Do **Open addressing with linear probing** *project (due Monday, Dec 1)*

Due **Fri, Nov 21** *(end of day)*
Read Section 7.3
Do Exercises 7.(4,5,7,8)
Take quiz

Due **Mon, Dec 1** *(but recommended before break)*
Read Sections 7.(4 & 5)
(No exercises or quiz)