

Chapter 6, Hash tables:

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

Today:

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

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

Uniformity:

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

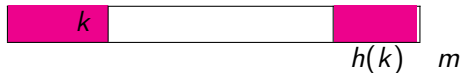
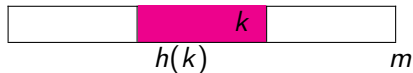
Independence:

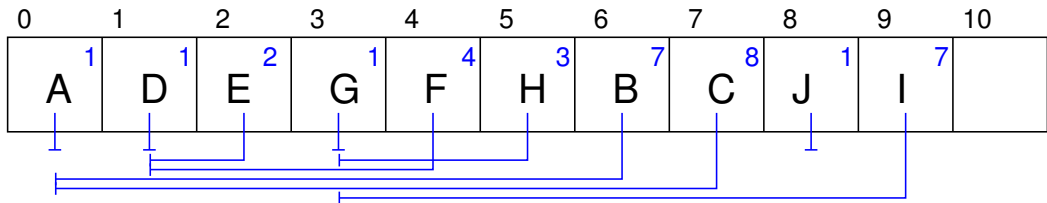
$$P(h(k_1) = i) = P(h(k_1) = i \mid h(k_2) = j)$$

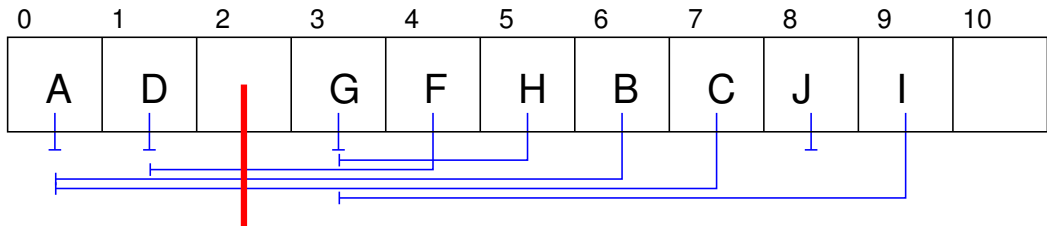


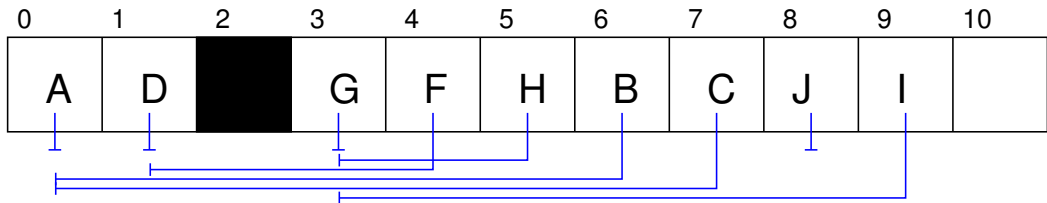
Invariant (Class OpenAddressingHashMap)

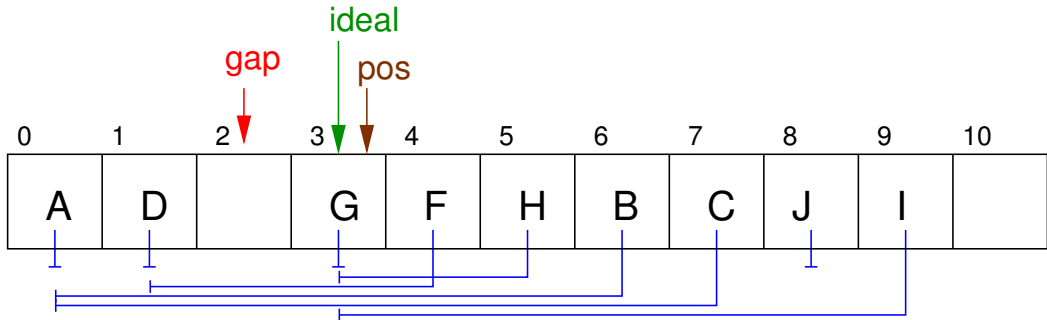
1. The table is not full; there exists $i \in [0, m)$ such that $\text{table}[i] = \text{null}$.
2. There are no breaks in the chain for any key in the table; for all $i \in [0, m)$ such that $\text{table}[i]$ contains key k ,
 - ▶ if $h(k) \leq i$, then for all $j \in [h(k), i]$, $\text{table}[j] \neq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

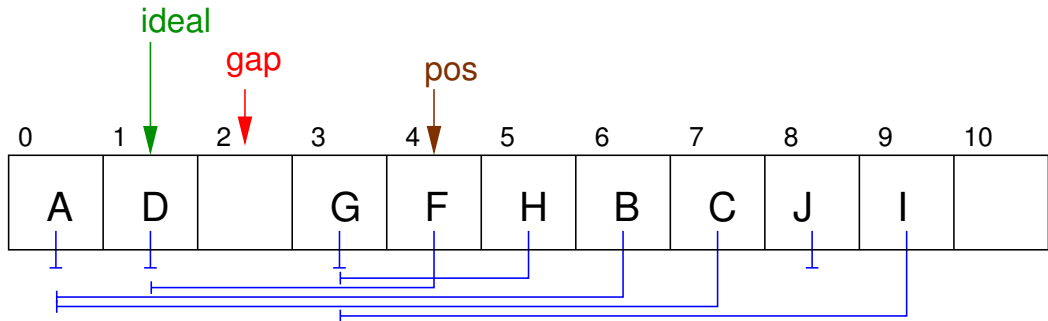


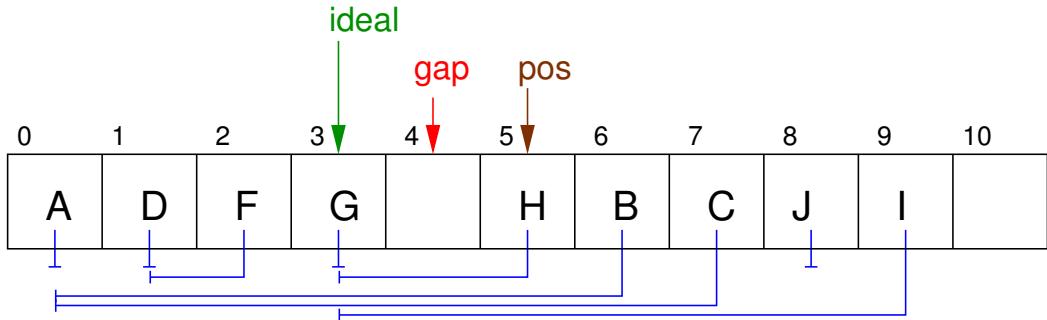


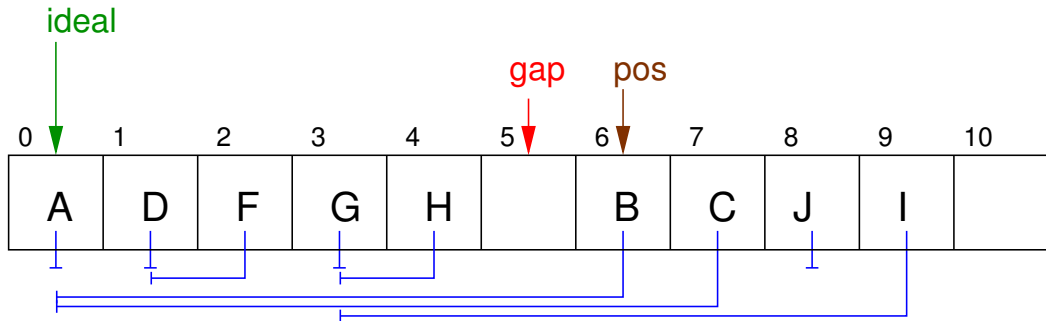


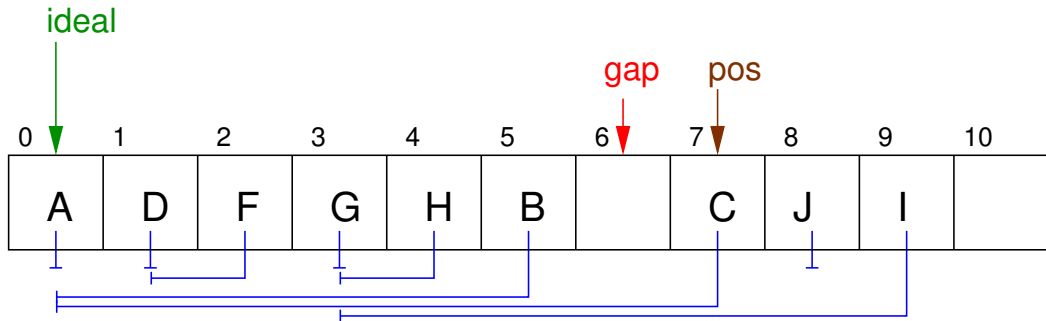


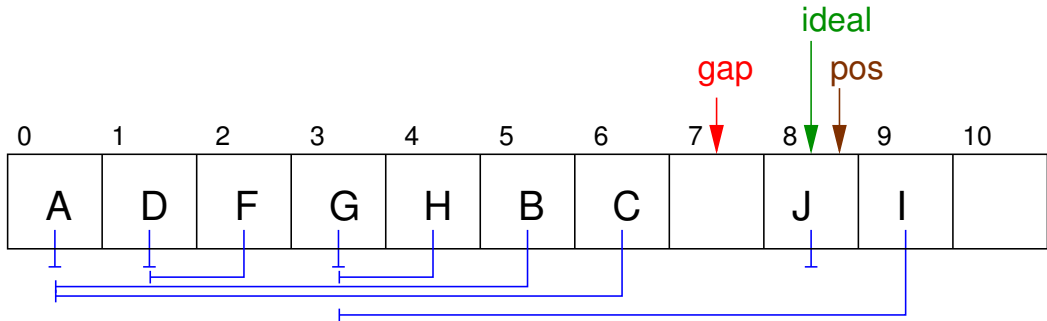


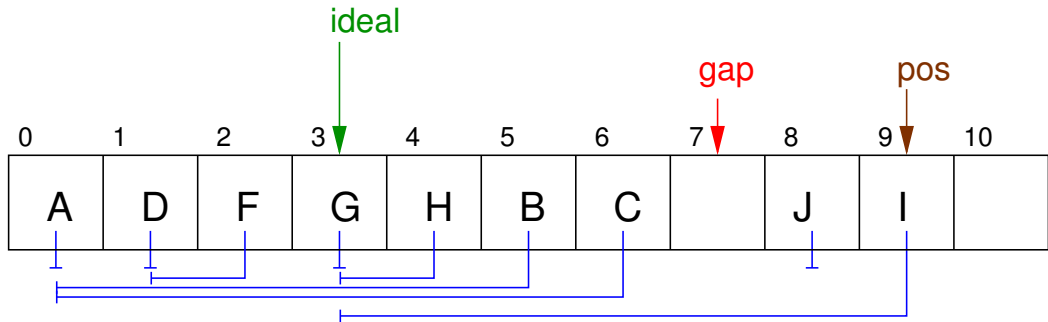


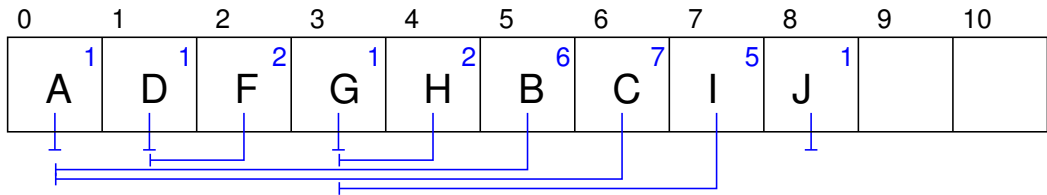




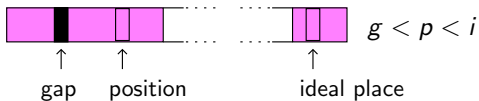
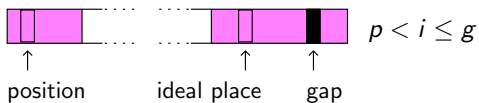
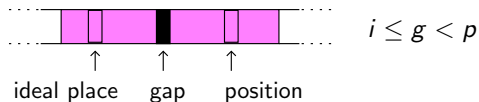




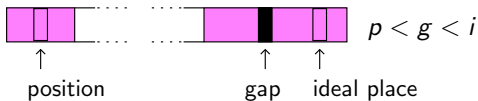
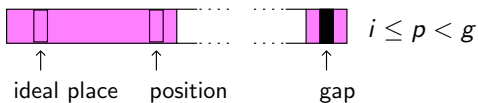
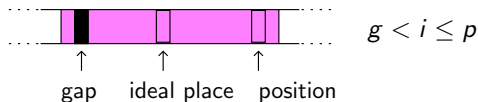




Cases to plug the gap



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 ($h(\text{keys}[k])$) and its actual place (k).

Coming up: *(all end-of-day)*

Do **Optimal BST** project *(due this past Monday, April 8)*

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

Due Thurs Apr 11

Read Section 7.3

Do Exercises 7.(4,5,7,8)

Take quiz

Due Mon, Apr 15

Read Sections 7.(4 & 5)

(No practice problems or quiz)