Chapter 6, Hash tables:

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

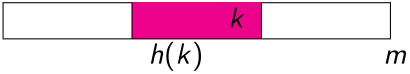
Today:

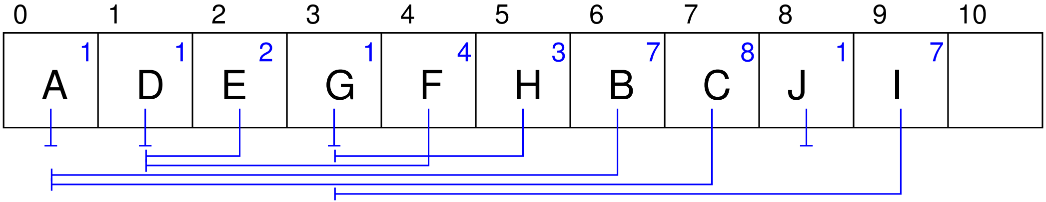
- Review/finish hash table concepts
- Basic idea and example of open addressing
- Terminology, code, and invariant
- Probing strategies
- Deletion

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

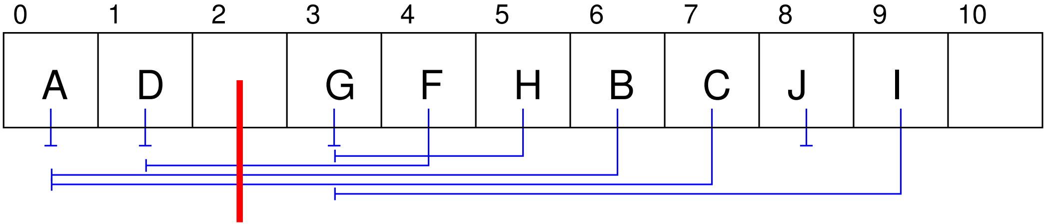
Invariant (Class `OpenAddressingHashMap`)

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* $\texttt{table}[i]$ *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* $j \in [0, i] \cup [h(k), m), \texttt{table}[j] \neq \texttt{null}$.

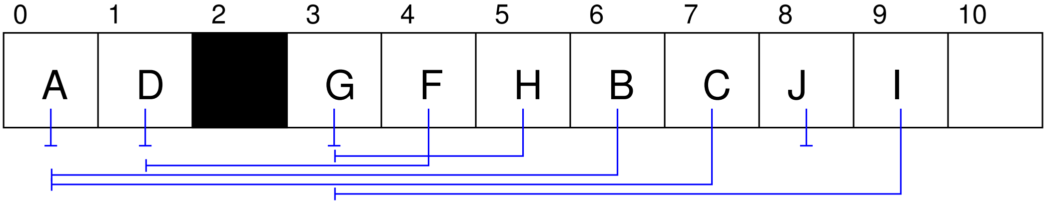


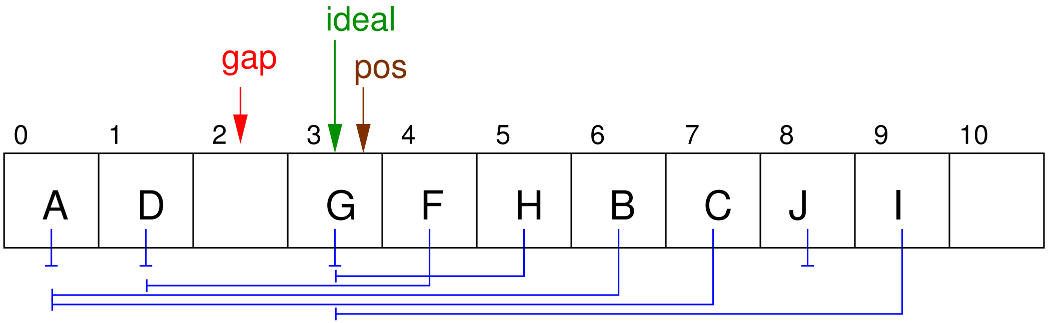


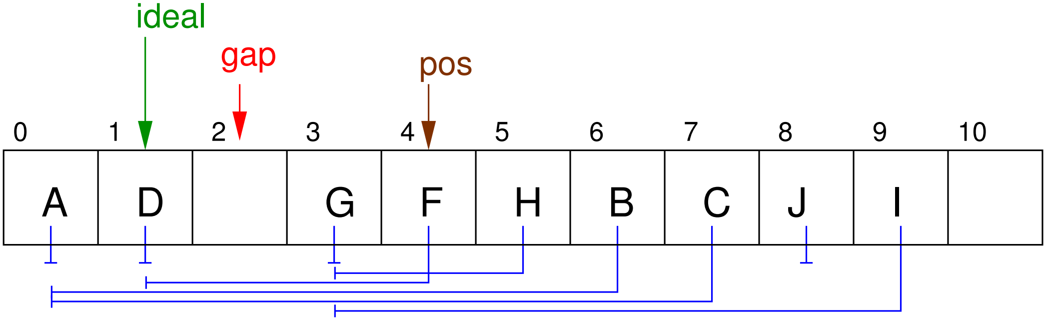
0 1 2 3 4 5 6 7 8 9 10
A 1
D 1
E 2
G 1
F 4
H 3
B 7
C 8
J 1
I 7

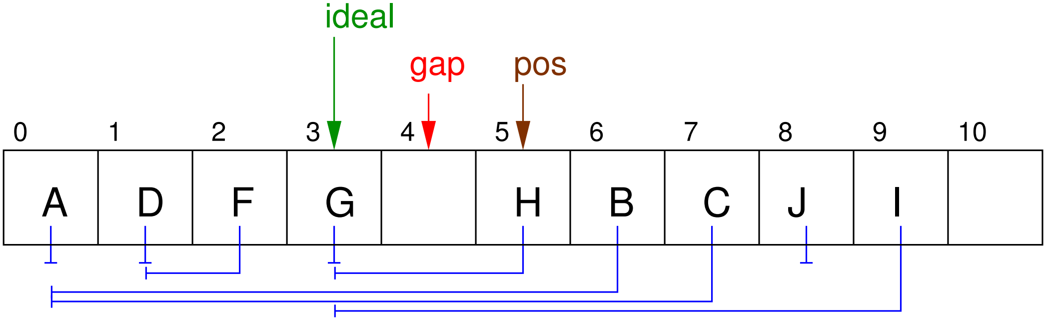


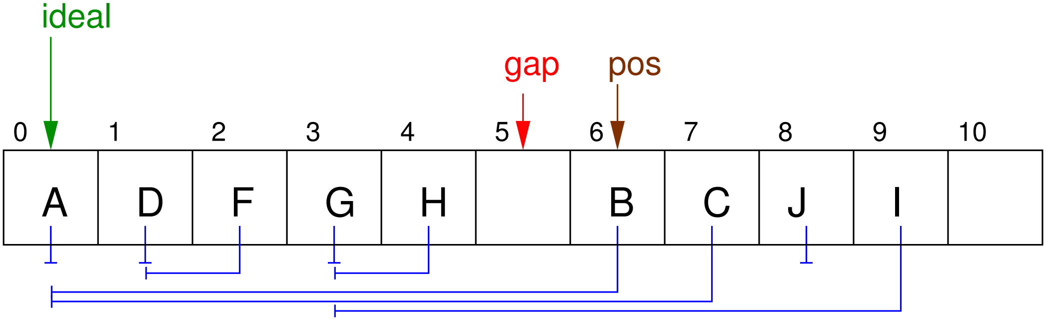
0
1
2
3
4
5
6
7
8
9
10
A
D
G
F
H
B
C
J
I


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

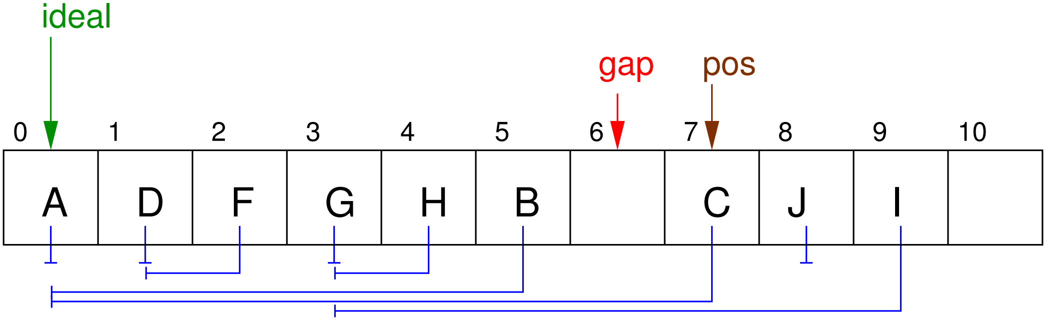

ideal
gap
pos
0
1
2
3
4
5
6
7
8
9
10
A
D
G
F
H
B
C
J
I


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

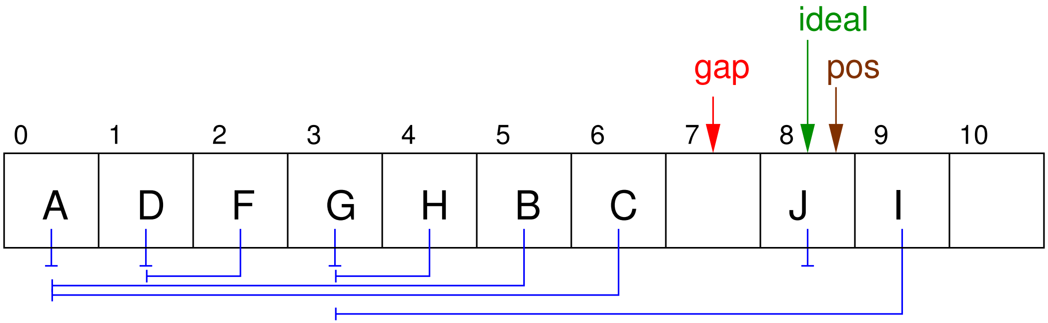

ideal
gap
pos
0
1
2
3
4
5
6
7
8
9
10
A
D
F
G
H
B
C
J
I

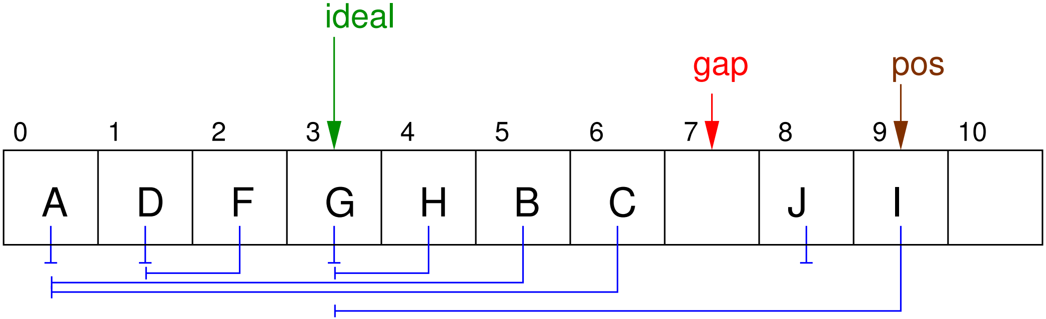


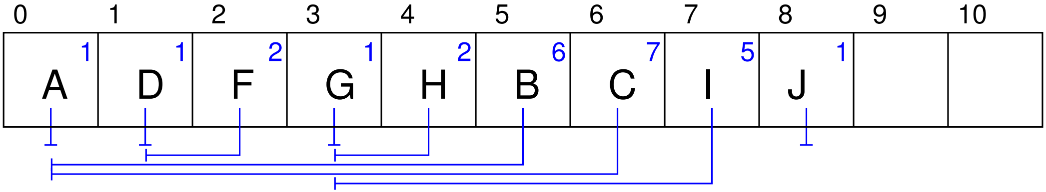
ideal
gap
pos
0
1
2
3
4
5
6
7
8
9
10
A
D
F
G
H
B
C
J
I

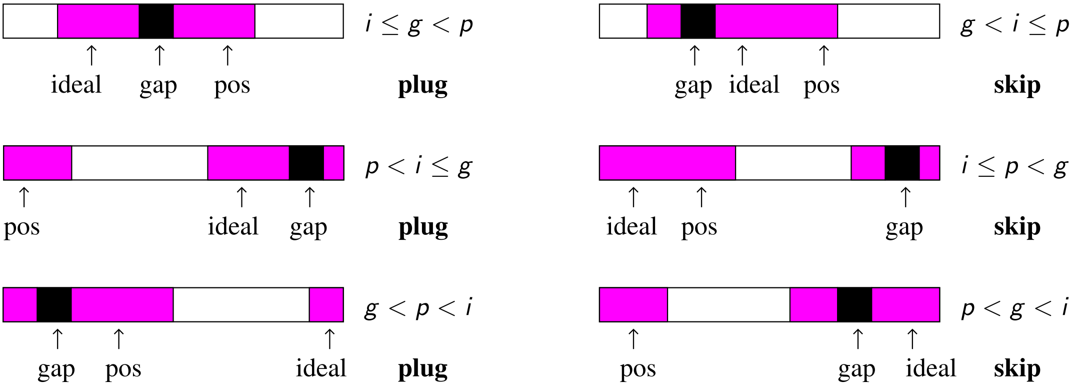


ideal
gap
pos
0
1
2
3
4
5
6
7
8
9
10
A
D
F
G
H
B
C
J
I


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



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 (suggested by this past Friday, April 8)*
Do **Open addressing with linear probing** *project (suggested by Monday, Apr 18)*

Due **Tues, Apr 12**
Do practice problem, recreating separate chaining example
Read Section 7.3
Take quiz

Due **Mon, Apr 18**
Read Sections 7.(4 & 5)
(No practice problems or quiz)