CSC263 Tutorial #5

Hash tables

February 10, 2023

Things covered in this tutorial

- ⋆ What's a Hash table?
- ⋆ What are the two ways to address hash collisions covered in this course?

Hash Tables

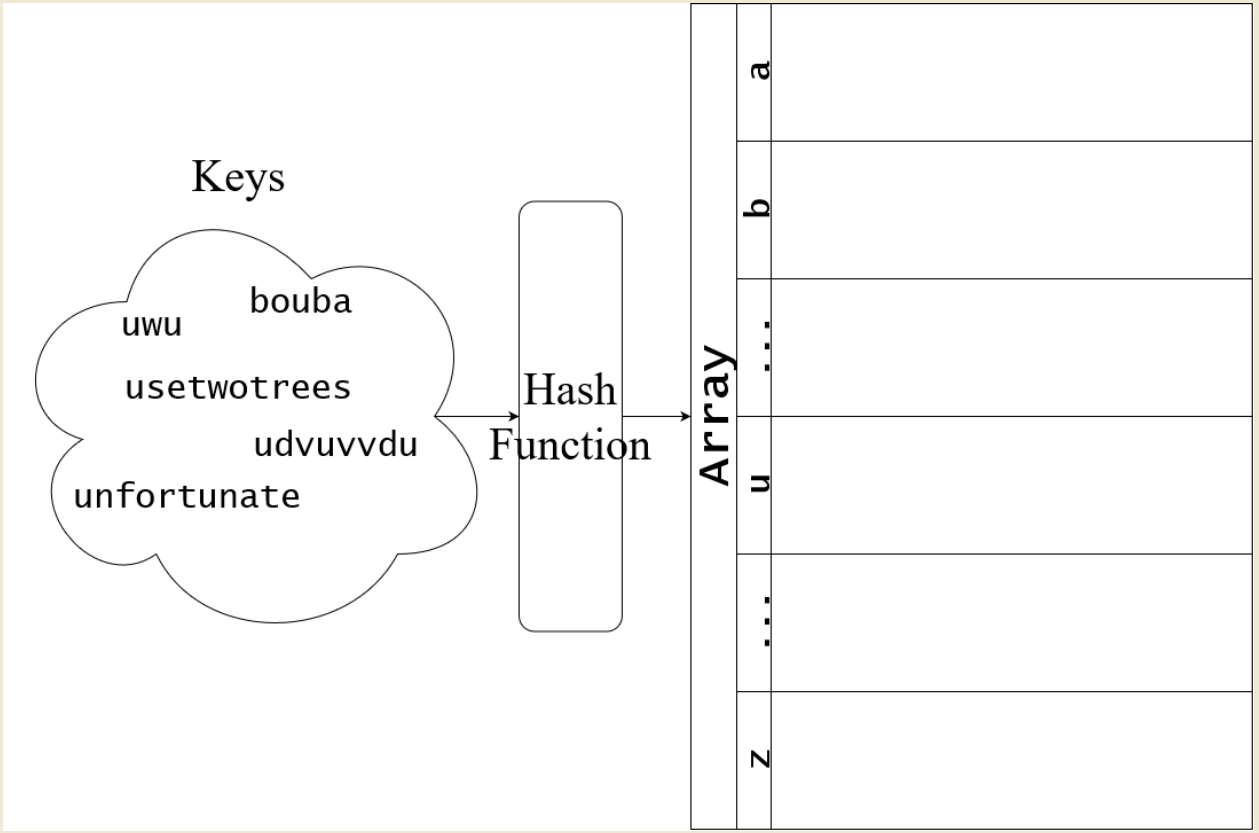
A hash table is an efficient way of implementing the Dictionary ADT.

Hash Tables

A hash table is an efficient way of implementing the Dictionary ADT.
It consists of an array of size m and a a *hash function*
$h : U \to \{0, \ldots, m-1\}$.

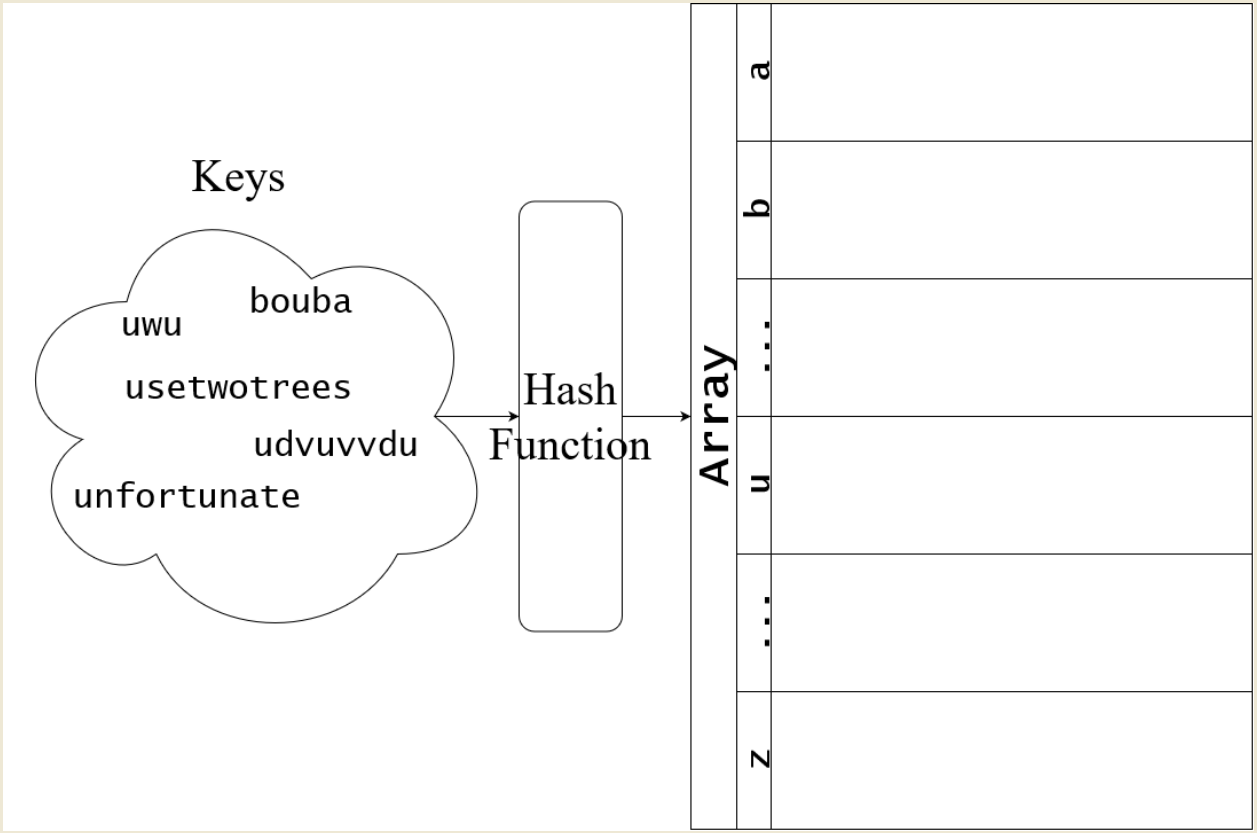
Hash Tables

A hash table is an efficient way of implementing the Dictionary ADT. It consists of an array of size m and a a *hash function* $h : U \to \{0, \ldots, m-1\}$.



Hash Tables

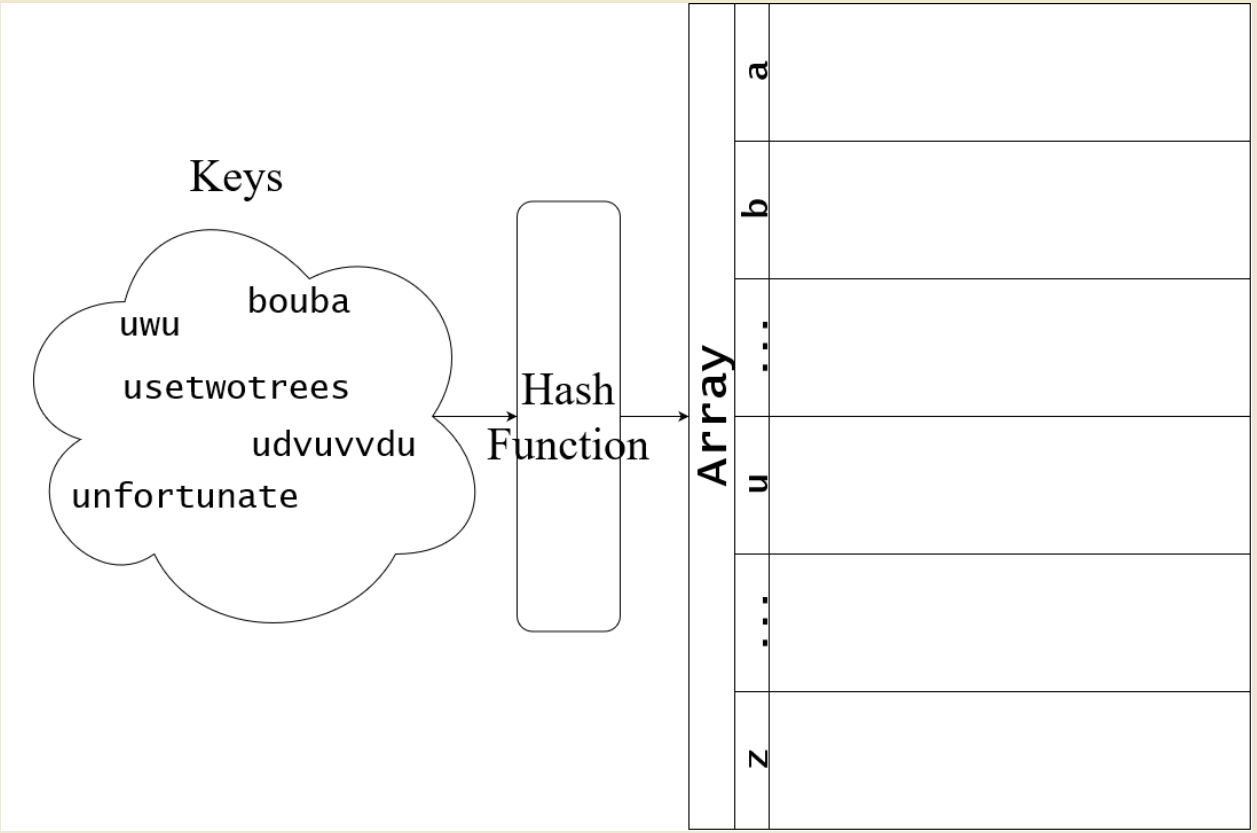
A hash table is an efficient way of implementing the Dictionary ADT. It consists of an array of size m and a a *hash function* $h : U \to \{0, \ldots, m-1\}$.



Problem: What if two distinct keys get mapped to the same address?

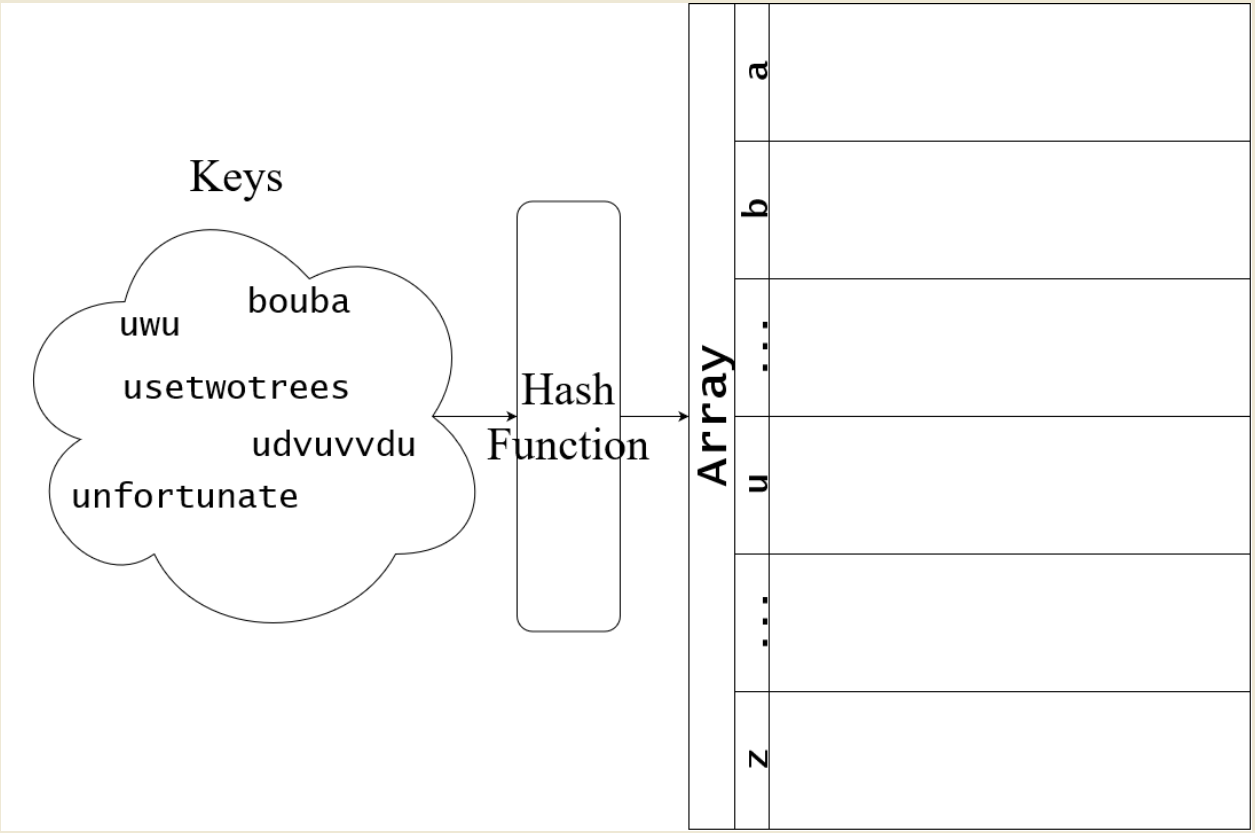
Hash Tables: Chaining

Chaining: Each array entry is a linked list.



Hash Tables: Chaining

Chaining: Each array entry is a linked list.



Task: Referring to the above diagram, assume the hash function is

$$h(w) = \text{first letter of } w.$$

Store all keys (in the "Keys" cloud) in the array.

Hash Tables: Chaining

Very fast![1] Average case:

- $\mathcal{O}(1)$ Insert.
- $\mathcal{O}(1)$ Delete.
- $\mathcal{O}(1)$ Search.

[1] Assuming the array size is large enough, that is...

Hash Tables: Chaining

Very fast![1] Average case:

- $\star$ $\mathcal{O}(1)$ Insert.
- $\star$ $\mathcal{O}(1)$ Delete.
- $\star$ $\mathcal{O}(1)$ Search.

[1] Assuming the array size is large enough, that is...

Hash Tables: Chaining

Worst case:

- $\star$ $\mathcal{O}(n)$ Insert.
- $\star$ $\mathcal{O}(n)$ Delete.
- $\star$ $\mathcal{O}(n)$ Search.

Hash Tables: Chaining

Worst case:

- $\star$ $\mathcal{O}(n)$ Insert.
- $\star$ $\mathcal{O}(n)$ Delete.
- $\star$ $\mathcal{O}(n)$ Search.

Hash Sort???

We've seen how heaps can implement sorting in $\mathcal{O}(n \log n)$ time.

Question: How would we sort with a hash table?

Hash Sort???

We've seen how heaps can implement sorting in $\mathcal{O}(n \log n)$ time.

Question: How would we sort with a hash table?

Answer: Don't try it; this is a terrible idea. Why not?

Tutorial Activity: Try Question 1(a) from the tutorial activities!

Hash Tables 2: Electric Boogaloo

Open addressing or probing is another way of resolving hash collisions.

Hash Tables 2: Electric Boogaloo

Open addressing or probing is another way of resolving hash collisions. For example, linear probing:

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

Hash Tables 2: Electric Boogaloo

Open addressing or probing is another way of resolving hash collisions. For example, linear probing:

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

Task: Suppose $m = 8$ and $h'(k) = k \bmod 8$. Insert $10, 22, 31, 4, 15, 28, 17$ into the hash table.

Hash Tables 2: Electric Boogaloo

Open addressing or probing is another way of resolving hash collisions. For example, linear probing:

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

Task: Suppose $m = 8$ and $h'(k) = k \bmod 8$. Insert $10, 22, 31, 4, 15, 28, 17$ into the hash table.

Hash Tables 2: Electric Boogaloo

Open addressing or probing is another way of resolving hash collisions. For example, linear probing:

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

Task: Suppose $m = 8$ and $h'(k) = k \bmod 8$. Insert $10, 22, 31, 4, 15, 28, 17$ into the hash table.

		10				22	
0	1	2	3	4	5	6	7

Hash Tables 2: Electric Boogaloo

Open addressing or probing is another way of resolving hash collisions. For example, linear probing:

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

Task: Suppose $m = 8$ and $h'(k) = k \bmod 8$. Insert $10, 22, 31, 4, 15, 28, 17$ into the hash table.

		10				22	31
0	1	2	3	4	5	6	7

Hash Tables 2: Electric Boogaloo

Open addressing or probing is another way of resolving hash collisions. For example, linear probing:

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

Task: Suppose $m = 8$ and $h'(k) = k \bmod 8$. Insert $10, 22, 31, 4, 15, 28, 17$ into the hash table.

		10		4		22	31
0	1	2	3	4	5	6	7

Hash Tables 2: Electric Boogaloo

Open addressing or probing is another way of resolving hash collisions. For example, linear probing:

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

Task: Suppose $m = 8$ and $h'(k) = k \bmod 8$. Insert $10, 22, 31, 4, 15, 28, 17$ into the hash table.

15		10		4		22	31
0	1	2	3	4	5	6	7

Hash Tables 2: Electric Boogaloo

Open addressing or probing is another way of resolving hash collisions. For example, linear probing:

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

Task: Suppose $m = 8$ and $h'(k) = k \bmod 8$. Insert $10, 22, 31, 4, 15, 28, 17$ into the hash table.

15		10		4	28	22	31
0	1	2	3	4	5	6	7

Hash Tables 2: Electric Boogaloo

Open addressing or probing is another way of resolving hash collisions. For example, linear probing:

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

Task: Suppose $m = 8$ and $h'(k) = k \bmod 8$. Insert $10, 22, 31, 4, 15, 28, 17$ into the hash table.

15		10	17	4	28	22	31
0	1	2	3	4	5	6	7