CPSC 420 Lecture 28: Today's announcements:

- HW4 is on Gradescope, due Mar 30, 23:59
- Examlet 4 on April 5 in class. Closed book & no notes
- Reading: Ch.5 Hash Tables [Director's Cut by Erickson]
- Reading: Cuckoo Hashing for Undergraduates [by Pagh]

Today's Plan

- Cuckoo Hashing

Cuckoo Hashing

Time per operation

Find	$O(1)$ time worst case
Delete	$O(1)$ time worst case
Insert	$O(1)$ expected, amortized time

How is this possible?

Cuckoo Hashing

Time per operation

Find	$O(1)$ time worst case
Delete	$O(1)$ time worst case
Insert	$O(1)$ expected, amortized time

- Use two hash functions h_1 and h_2.
- Item x will be stored in slot $h_1(x)$ or $h_2(x)$ of hash table T.
- Each slot in the hash table can contain at most one item.
- $n =$ maximum number of items stored at any time
- $m =$ size of hash table T $(m > n)$

Cuckoo Hashing

Time per operation

Find	$O(1)$ time worst case
Delete	$O(1)$ time worst case
Insert	$O(1)$ expected, amortized time

- Use two hash functions h_1 and h_2.
- Item x will be stored in slot $h_1(x)$ or $h_2(x)$ of hash table T.
- Each slot in the hash table can contain at most one item.
- $n =$ maximum number of items stored at any time
- $m =$ size of hash table T $(m > n)$

On an **insert**(x) collision, item x kicks the resident item y out. Item y then goes to its alternate slot (kicking whoever's there out). Etc. Etc.

Cuckoo Hashing


[1]

[1] Per Harald Olsen (Wikipedia)

Insert Algorithm

insert(x)

1. if $T[h_1(x)] = x$ or $T[h_2(x)] = x$ return
2. $i \leftarrow h_1(x)$
3. repeat n times
4. $\quad y \leftarrow T[i]$
5. $\quad T[i] \leftarrow x$ — put x into slot i
6. $\quad$ if $y =$ NULL return — done
7. $\quad$ if $i = h_1(y)$ then $\underbrace{i \leftarrow h_2(y)}$ else $\underbrace{i \leftarrow h_1(y)}$
8. $\quad x \leftarrow y$
9. rehash; insert(x)

i : alternate slot for y

Example and Cuckoo Graph

items

	h_1	h_2
A	0	3
B	3	8
C	1	2
H	4	7
P	5	8
W	7	4
X	7	0
Y	3	4

$n = 8$

hash table

	T
0	A
1	
2	
3	
4	
5	
6	
7	
8	
$m - 1 = 9$	

A

Example and Cuckoo Graph

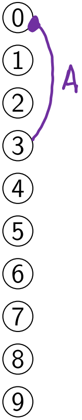
hash table

	h_1	h_2
A	0	3
B	3	8
C	1	2
H	4	7
P	5	8
W	7	4
X	7	0
Y	3	4

$n = 8$

	T
0	A
1	
2	
3	
4	
5	
6	
7	
8	
$m-1=9$	

cuckoo graph



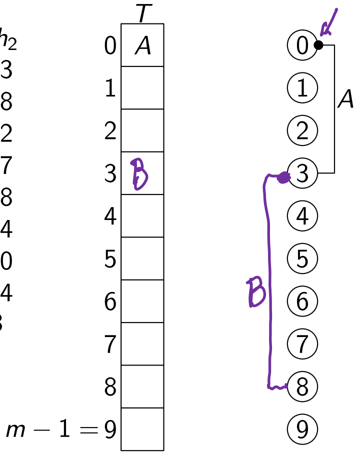
Vertex i for every slot $i = 0, 1, \ldots, m-1$.

Edge $(h_1(x), h_2(x))$ for every item x.

Example and Cuckoo Graph

	h_1	h_2
A	0	3
B	3	8
C	1	2
H	4	7
P	5	8
W	7	4
X	7	0
Y	3	4

$n = 8$

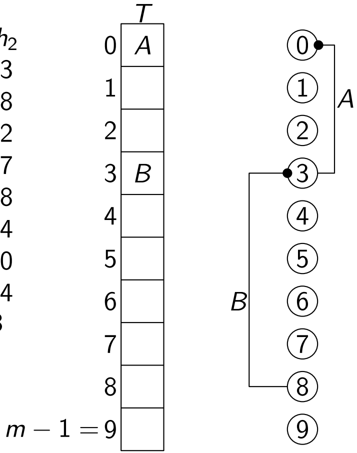


Vertex i for every slot $i = 0, 1, \ldots, m-1$.
Edge $(h_1(x), h_2(x))$ for every item x.

Example and Cuckoo Graph

	h_1	h_2
✓ A	0	3
✓ B	3	8
C	1	2
H	4	7
P	5	8
W	7	4
X	7	0
Y	3	4

$n = 8$

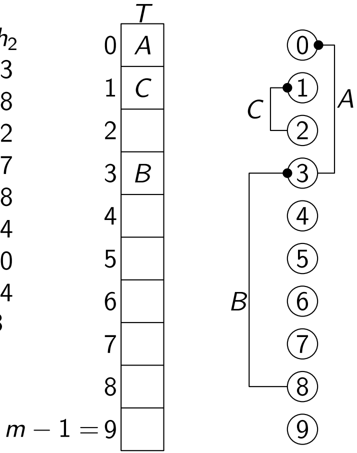


Vertex i for every slot $i = 0, 1, \ldots, m-1$.
Edge $(h_1(x), h_2(x))$ for every item x.

Example and Cuckoo Graph

	h_1	h_2
A	0	3
B	3	8
C	1	2
H	4	7
P	5	8
W	7	4
X	7	0
Y	3	4

$n = 8$



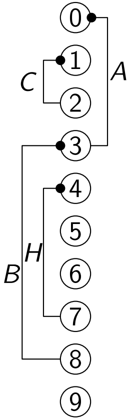
Vertex i for every slot $i = 0, 1, \ldots, m-1$.
Edge $(h_1(x), h_2(x))$ for every item x.

Example and Cuckoo Graph

	h_1	h_2
A	0	3
B	3	8
C	1	2
H	4	7
P	5	8
W	7	4
X	7	0
Y	3	4

$n = 8$

	T
0	A
1	C
2	
3	B
4	H
5	
6	
7	
8	
$m-1=9$	



Vertex i for every slot $i = 0, 1, \ldots, m-1$.

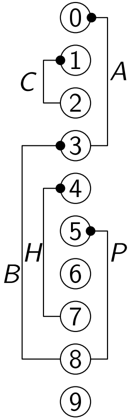
Edge $(h_1(x), h_2(x))$ for every item x.

Example and Cuckoo Graph

	h_1	h_2
A	0	3
B	3	8
C	1	2
H	4	7
P	5	8
W	7	4
X	7	0
Y	3	4

$n = 8$

	T
0	A
1	C
2	
3	B
4	H
5	P
6	
7	
8	
$m-1=9$	

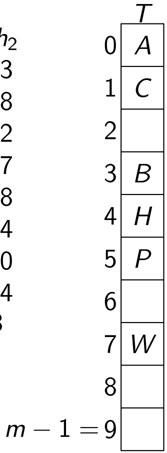


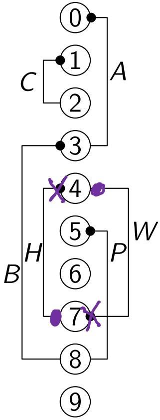
Vertex i for every slot $i = 0, 1, \ldots, m-1$.
Edge $(h_1(x), h_2(x))$ for every item x.

Example and Cuckoo Graph

	h_1	h_2
A	0	3
B	3	8
C	1	2
H	4	7
P	5	8
W	7	4
X	7	0
Y	3	4

$n = 8$





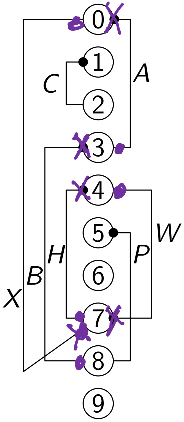
Vertex i for every slot $i = 0, 1, \ldots, m-1$.
Edge $(h_1(x), h_2(x))$ for every item x.

Example and Cuckoo Graph

	h_1	h_2
A	0	3
B	3	8
C	1	2
H	4	7
P	5	8
W	7	4
X	7	0
Y	3	4

$n = 8$

	T
0	A
1	C
2	
3	B
4	H
5	P
6	
7	W
8	
$m - 1 = 9$	



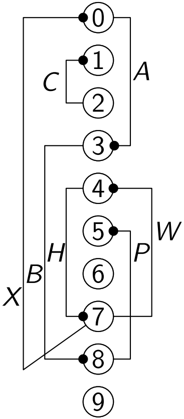
Vertex i for every slot $i = 0, 1, \ldots, m - 1$.
Edge $(h_1(x), h_2(x))$ for every item x.

Example and Cuckoo Graph

	h_1	h_2
A	0	3
B	3	8
C	1	2
H	4	7
P	5	8
W	7	4
X	7	0
Y	3	4

$n = 8$

	T
0	X
1	C
2	
3	A
4	W
5	P
6	
7	H
8	B
$m-1=9$	



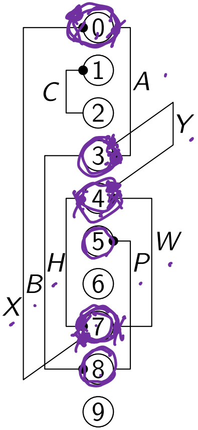
Vertex i for every slot $i = 0, 1, \ldots, m-1$.
Edge $(h_1(x), h_2(x))$ for every item x.

Example and Cuckoo Graph

	h_1	h_2
A	0	3
B	3	8
C	1	2
H	4	7
P	5	8
W	7	4
X	7	0
Y	3	4

$n = 8$

	T
0	X
1	C
2	
3	A
4	W
5	P
6	
7	H
8	B
$m-1=9$	

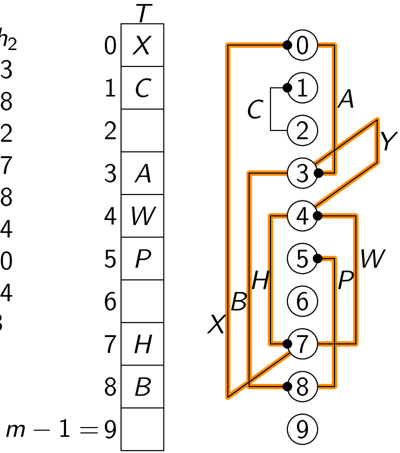


Vertex i for every slot $i = 0, 1, \ldots, m-1$.
Edge $(h_1(x), h_2(x))$ for every item x.

Example and Cuckoo Graph

	h_1	h_2
A	0	3
B	3	8
C	1	2
H	4	7
P	5	8
W	7	4
X	7	0
Y	3	4

$n = 8$



connected component in cuckoo graph has more edges than vertices

then NO ROOM for all items in component

Vertex i for every slot $i = 0, 1, \ldots, m-1$.

Edge $(h_1(x), h_2(x))$ for every item x.

Cuckoo Insert Analysis

Pagh

insert(x) only visits slots that are connected in the cuckoo graph to $h_1(x)$ or $h_2(x)$.

We say x and y are in the same **bucket** if there is a path in the cuckoo graph from $h_1(x)$ or $h_2(x)$ to $h_1(y)$ or $h_2(y)$.

Only elements in the same bucket as x can impact the runtime of **insert**(x).

Lemma 2: The probability that x and y are in the same bucket is $O(1/m)$.

To show this we first prove:

2   # slots in table   max # items we hash

Lemma 1: For any slots i and j and any $c > 1$, if $m \geq 2cn$ then the probability that a shortest path from i to j has length ℓ in the cuckoo graph is $\leq \frac{1}{c^{\ell} \cancel{r}\, m}$.

Cuckoo Insert Analysis

Lemma 1: For any slots i and j and any $c > 1$, if $m \geq 2cn$ then the probability that a shortest path from i to j has length ℓ in the cuckoo graph is $\leq \frac{1}{c^\ell m}$.

Proof: Such a path exists of length $\ell = 1$ iff some item x has $(h_1(x), h_2(x)) = (i, j)$ or (j, i). This happens with probability $\leq \underset{\text{(choices for } x)}{n} (2/m^2) \leq 1/(cm)$ (assuming h_1 and h_2 are random).

Proceed by induction on ℓ. There is a shortest path from i to j of length $\boxed{\ell \geq 2}$ iff there is
(1) a shortest path of length $\ell - 1$ from i to k (that avoids j)

$\boxed{\text{probability} \leq \frac{1}{c^{\ell-1} m}}$ by induction

and (2) an edge from k to j (for some $k \neq i, j$).

probability $\leq n(2/m^2) \leq \boxed{1/(cm)}$

Probability of (1) and (2) $\leq \underset{\text{(choices for } k)}{m} \frac{1}{c^\ell m^2} = \frac{1}{c^\ell m}$. □

Cuckoo Insert Analysis

Lemma 2: The probability that x and y are in the same bucket is $O(1/m)$.

i j

Proof: Items x and y are in the same bucket iff there is a path of length ℓ (for some ℓ) from $h_1(x)$ or $h_2(x)$ to $h_1(y)$ or $h_2(y)$. This happens with probability $\leq 4\sum_{\ell=1}^{\infty} \frac{1}{c^{\ell} m} = \frac{4}{(c-1)m} = O(1/m)$. □

Theorem: If there is no cycle in the cuckoo graph then the expected time for insert(x) is $O(1)$

Proof: Only those items $y \neq x$ that are in the same bucket as x can cause cuckoo swaps during insert(x) and each y causes at most one swap (assuming there is no cycle). The probability that item y is in the same bucket as x is $O(1/m)$ (Lemma 2). So the total expected number of swaps is $\leq \underbrace{(n-1)}_{\text{(choices for } y)} \cdot O(1/m) = O(1)$ (since $n < m$). □

Cuckoo Rehash

```
insert(x)
if T[h_1(x)] = x or T[h_2(x)] = x return
i ← h_1(x)
repeat n times
        y ← T[i]
        T[i] ← x
        if y = NULL return
        if i = h_1(y) then i ← h_2(y) else i ← h_1(y)
        x ← y
rehash; insert(x)
```

Lemma 3: If $m \geq 2cn$ then the probability of a cycle in the cuckoo graph after n insertions is at most $\frac{1}{c-1}$.

Proof: Slot i is involved in a cycle iff there is a path from i to itself of length $\ell \geq 1$. By Lemma 1, this happens with probability $\leq \sum_{\ell=1}^{\infty} \frac{1}{c^{\ell} m} = \frac{1}{(c-1)m}$. Summing over all m slots, gives probability $\leq \frac{1}{c-1}$ for a cycle. $\square$

Cuckoo Rehash

Lemma 3: If $m \geq 2cn$ then the probability of a cycle in the cuckoo graph after n insertions is at most $\frac{1}{c-1}$.

Proof: Slot i is involved in a cycle iff there is a path from i to itself of length $\ell \geq 1$. By Lemma 1, this happens with probability $\leq \sum_{\ell=1}^{\infty} \frac{1}{c^\ell m} = \frac{1}{(c-1)m}$. Summing over all m slots, gives probability $\leq \frac{1}{c-1}$ for a cycle. □

Each rehash takes $O(n)$ time.

By Lemma 3, for $c > 3$, the prob. that one rehash occurs after n insertions is $\leq 1/2$, that two rehashes occur $\leq 1/4$, etc. So expected amortized cost of rehash is $O(1)$.

Note: A rehash triggers $k > 0$ consecutive rehashes with prob. $\leq 1/2^k$. So the expected cost is still $O(n) \cdot \sum_{k=1}^{\infty} 1/2^k = O(n)$.