- Framework is out - due by midnight on 12/1

- Project due tomorrow

Announcement

CS 180 - Fresh Tables 4.3
- Flight # 4 arrival 1:30

- locker number on name

Examples:

```c
long find (keyType K);
```

```c
void insert (keyType K); dataType R;
```

A structure which supports the following:

Dictionaries:
Hashing - big picture
Hashing for fast lookups

Many to 1
Not

Cell's can

1

1

OUT

OUT

OUT

OUT

OUT

OUT

OUT

OUT

OUT
Key space is larger than our

No

Can we ever perfectly avoid collisions?
How can we handle collisions?

(Do we have data structures to store more than one thing?)
Insert 2

Ex: 2

1

2

3

4

5

6

Left child (38)

36 37 38

40 41 42
(Assuming the vector fills up)

-2

until it's empty. Just keep checking next spot instead of lists. If we reach to a full linear problem.
\(15 \mod 11 = 4\)
\(3 \mod 11 = 3\)
\(5 \mod 11 = 5\)
\(7 \mod 11 = 7\)
\(19 \mod 11 = 8\)
\(6 \mod 11 = 6\)
\(26 \mod 11 = 4\)
\(37 \mod 11 = 4\)
\(21 \mod 11 = 10\)
\(12 \mod 11 = 1\)

- Map is \(h(x) = x \mod 11\)

\((x, y) = (13, 4)\)

Example: Use Linear probing
Usually, treat a short-term model on $O(n^3)$.

Removal. Ignored.

Isn't it?

$O(n)$ is $n$ times inserted.

Running things.

$N = \text{size of table}$
\[ A \left[ \frac{h(i)+4}{N} \mod N \right] \] 
\[ A \left[ \frac{h(i)+4}{N} \mod N \right] \]

where \( j = 0, 1, 2, 3, \ldots \)

To avoid clusters, instead my

\[ A \left[ \frac{h(i)+j}{N} \mod N \right] \] 
\[ A \left[ \frac{h(i)+j}{N} \mod N \right] \]

Notice: Linear probing doesn't work.

Quadratic Probing.
My actually feel is half full, my Feel if any

Even with

A need to be grown by this

Still cause "Second climbing"

A quick fix? Problem issue?
Given \( h'(x) \) is a function, \( f(x) = x \cdot h'(x) \)

\[ f'(x) = \frac{\partial}{\partial x} \left[ x \cdot h'(x) \right] \]

Try \( A(h(x)) \) for linear hashing?

Double Hashing
A lot of code periodically checks if $n$ is greater than $N$. If it is, it replaces $n$ with $N$.

Even chaining gets worse.

Most of these techniques only work well if $N/n > 5$.

Good Factors