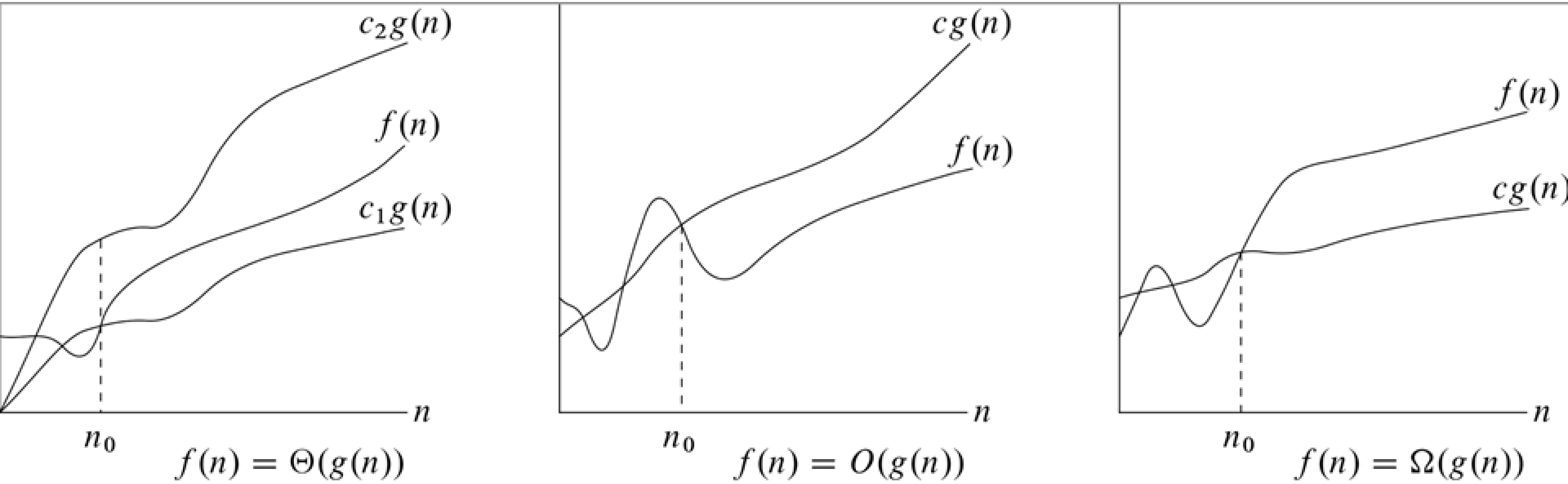
Lecture 2: Limit Tests and ADTs

These slides include material originally prepared by Dr.Ron Cytron, Dr. Jeremy Buhler, and Dr. Steve Cole.

Announcements

- Lab 1 is due on 2/8
  - Log in to Gradescope and try uploading something soon! Don't get caught by tech issue at last minute.
- Studio 2 Thursday (do short pre-lab beforehand)
- Office Hours locations now posted
- Outline for today:
  - Limit Tests
  - Algorithm Comparisons
  - Abstract Data Types
  - Linked Lists

Review from Last Time: O, Ω, Θ



Describes *order of growth* ignoring behavior for small n, constant factors

How Did We Prove O, Ω, and Θ Bounds?

- Pick constants c, n_0 (two constants needed for Θ)
- Prove the right bounding inequality (or pair of inequalities for Θ)
- Used a variety of math tools for this proof
  - arithmetic
  - algebra
  - calculus

How Did We Prove O, Ω, and Θ Bounds?

- **Pick constants c, n_0 (two constants needed for Θ)**
- **Prove the right bounding inequality (or pair of inequalities for Θ)**
- **Used a variety of math tools for this proof**
  - **arithmetic**
  - **algebra**
  - **calculus**

Is there a single, uniform proof strategy that gives answers quickly and requires less creativity?

Let's Talk About Limits

Suppose $$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$$

Let's Talk About Limits

Suppose $$\lim_{n \to \infty} \frac{f(n)}{g(n)} = 0$$

Which function grows faster?

Let's Talk About Limits

Suppose $$\lim_{n \to \infty} \frac{f(n)}{g(n)} = 0$$

Which function grows faster? g(n)

Let’s Talk About Limits

Suppose

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$$

Which function grows faster? g(n)

Claim: f(n) = O(g(n))

Let's Talk About Limits

Suppose $$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$$

Which function grows faster? g(n)

Claim: f(n) = O(g(n))

For *any constant c > 0*, we can find n_0 s.t.

$f(n)/g(n) \leq c$ when $n \geq n_0$.

Defn of Limit

Let’s Talk About Limits

Suppose $$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$$

Which function grows faster? g(n)

Claim: $f(n) = O(g(n))$

For $c=1$, we can find n_0 s.t.

$f(n) \leq c\, g(n)$ when $n \geq n_0$.

Let's Talk About Limits

Suppose $$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$$

Which function grows faster? g(n)

Claim: f(n) = O(g(n))

For c=1, we can find n_0 s.t.

$f(n) \le c\, g(n)$ when $n \ge n_0$. QED

Defn of O(g(n))

Let's Talk About Limits

Suppose $$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$$

Which function grows faster? g(n)

Claim: f(n) ≠ Ω(g(n))

Let’s Talk About Limits

Suppose $$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$$

Which function grows faster? g(n)

Claim: f(n) ≠ Ω(g(n))

Suppose f(n) = Ω(g(n)).

Let's Talk About Limits

Suppose $$\lim_{n \to \infty} \frac{f(n)}{g(n)} = 0$$

Which function grows faster? g(n)

Claim: $f(n) \neq \Omega(g(n))$

Suppose $f(n) = \Omega(g(n))$. Then for some $c > 0$, $n_0 > 0$,

$$f(n) \geq c\, g(n) \text{ when } n \geq n_0.$$

Let's Talk About Limits

Suppose $$\lim_{n \to \infty} \frac{f(n)}{g(n)} = 0$$

Which function grows faster? g(n)

Claim: $f(n) \neq \Omega(g(n))$

Suppose $f(n) = \Omega(g(n))$. Then for some $c > 0$, $n_0 > 0$,

$$f(n)/g(n) \geq c \text{ when } n \geq n_0.$$

Let's Talk About Limits

Suppose $$\lim_{n \to \infty} \frac{f(n)}{g(n)} = 0$$

Which function grows faster? g(n)

Claim: $f(n) \neq \Omega(g(n))$

Suppose $f(n) = \Omega(g(n))$. Then for some $c > 0$, $n_0 > 0$,

But then the limit above would be at least $c > 0$.

Let's Tall

Contradiction!

Suppose

Which fun

Claim: f(n)

Suppose f > 0,

Let's Talk About Limits

Suppose $$\lim_{n \to \infty} \frac{f(n)}{g(n)} = 0$$

Which function grows faster? g(n)

Claim: f(n) ≠ Ω(g(n))

Suppose f(n) = Ω(g(n)). ⟵ **Implication: this assumption was false. QED**

What Have We Learned?

- **Thm**: when $$\lim_{n \to \infty} \frac{f(n)}{g(n)} = 0$$
- f(n) = O(g(n)) but f(n) ≠ Ω(g(n))

What Have We Learned?

- **Thm**: when $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$
- f(n) = O(g(n)) but f(n) ≠ Ω(g(n))
- We sometimes write "f(n) = o(g(n))"
- "f(n) is little-o of g(n)"

More Limits…

Suppose $$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$$

Now which function grows faster?

More Limits…

Suppose $$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$$

Now which function grows faster? f(n)

More Limits…

Suppose $$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$$

Now which function grows faster? f(n)

Thm: f(n) = Ω(g(n)) but f(n) ≠ O(g(n))

[proof strategy is basically identical to previous]

More Limits…

Suppose $$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$$

Now which function grows faster? f(n)

Thm: f(n) is Ω(g(n)) but not O(g(n))

We sometimes write "f(n) = ω(g(n))"

"f(n) is little-omega of g(n)"

Yet More Limits

Suppose

$$\lim_{n \to \infty} \frac{f(n)}{g(n)} = k, \quad k > 0$$

Yet More Limits

Suppose

$$\lim_{n \to \infty} \frac{f(n)}{g(n)} = k, \quad k > 0$$

If the ratio converges to k, then after some point it must be ≤ k + ϵ for any fixed ϵ > 0.

Then

$$\exists n_0 \mid \forall n \geq n_0 \quad f(n) \leq (k + \epsilon) g(n) \rightarrow f(n) = O(g(n))$$

Yet More Limits

Suppose $$\lim_{n\to\infty} \frac{f(n)}{g(n)} = k, \quad k > 0$$

Similar logic: after some point the ratio must be ≥ k - ϵ for any fixed ϵ > 0.

Then

$$\exists n_0 \mid \forall n \geq n_0 \quad f(n) \leq (k+\epsilon)g(n) \rightarrow f(n) = O(g(n))$$

$$\exists n_0 \mid \forall n \geq n_0 \quad g(n) \leq \frac{f(n)}{(k-\epsilon)} \rightarrow f(n) = \Omega(g(n))$$

Yet More Limits

Suppose $$\lim_{n\to\infty} \frac{f(n)}{g(n)} = k, \quad k > 0$$

Thm: f(n) = Θ(g(n))

[we just proved this]

Summary: the Limit Test

If…

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$$

Then…

$$f(n) = O(g(n))$$

~~$f(n) = \Omega(g(n))$~~

If...	Then...	
$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$	$f(n) = O(g(n))$	~~$f(n) = \Omega(g(n))$~~
$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$	~~$f(n) = O(g(n))$~~	$f(n) = \Omega(g(n))$

If…	Then…	
$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$	$f(n) = O(g(n))$	~~$f(n) = \Omega(g(n))$~~
$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$	~~$f(n) = O(g(n))$~~	$f(n) = \Omega(g(n))$
$\lim_{n\to\infty} \frac{f(n)}{g(n)} = k, \;\; k > 0$	$f(n) = \Theta(g(n))$	

If…	Then…
$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$	**f(n) = o(g(n))**
$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$	**f(n) = ω(g(n))**
$\lim_{n\to\infty} \frac{f(n)}{g(n)} = k, \ \ k > 0$	**f(n) = Θ(g(n))**

If…	**Then…**
$$\lim_{n \to \infty} \frac{f(n)}{g(n)} = 0$$	**"f(n) grows slower than g(n)"**
$$\lim_{n \to \infty} \frac{f(n)}{g(n)} = \infty$$	**"f(n) grows faster than g(n)"**
$$\lim_{n \to \infty} \frac{f(n)}{g(n)} = k, \;\; k > 0$$	**"f(n), g(n) grow at same rate"**

If…

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$$

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$$

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = k, \;\; k > 0$$

Then…

"g(n) grows faster than f(n)"

These are *asymptotic* statements

"f(n) grows faster than g(n)"

"f(n), g(n) grow at same rate"

Examples

Compare n^3 with n^2

$$\lim_{n\to\infty} \frac{f(n)}{g(n)}$$

0	f(n) = o(g(n))
∞	f(n) = ω(g(n))
k > 0	f(n) = Θ(g(n))

$$\lim_{n\to\infty} \frac{n^3}{n^2}$$

Compare n^3 with n^2

$$\lim_{n\to\infty} \frac{f(n)}{g(n)}$$

0	**f(n) = o(g(n))**
∞	**f(n) = ω(g(n))**
k > 0	**f(n) = Θ(g(n))**

$$\lim_{n\to\infty} \frac{n^3}{n^2}$$

$$= \lim_{n\to\infty} n^1 = \infty$$

Compare n^3 with n^2

$$\lim_{n\to\infty} \frac{f(n)}{g(n)}$$

0	f(n) = o(g(n))
∞	f(n) = ω(g(n))
k > 0	f(n) = Θ(g(n))

$$\lim_{n\to\infty} \frac{n^3}{n^2}$$

$$= \lim_{n\to\infty} n^1 = \infty$$

$n^3 = \omega(n^2)$

Compare n^3 with n^2

$$\lim_{n\to\infty}\frac{f(n)}{g(n)}$$

0	$f(n) = o(g(n))$
∞	$f(n) = \omega(g(n))$
$k > 0$	$f(n) = \Theta(g(n))$

$$\lim_{n\to\infty}\frac{n^3}{n^2}$$

$$= \lim_{n\to\infty} n^1 = \infty$$

$n^3 = \Omega(n^2)$, but $n^3 \neq O(n^2)$

Compare log n with n

$$\lim_{n\to\infty} \frac{f(n)}{g(n)}$$

0	f(n) = o(g(n))
∞	f(n) = ω(g(n))
k > 0	f(n) = Θ(g(n))

$$\lim_{n\to\infty} \frac{\log n}{n} = ?$$

Compare log n with n

$$\lim_{n\to\infty} \frac{f(n)}{g(n)}$$

0	**f(n) = o(g(n))**
∞	**f(n) = ω(g(n))**
k > 0	**f(n) = Θ(g(n))**

$$\lim_{n\to\infty} \frac{\log n}{n} = ?$$

Undefined!

Compare log n with n

$$\lim_{n \to \infty} \frac{f(n)}{g(n)}$$

0	f(n) = o(g(n))
∞	f(n) = ω(g(n))
k > 0	f(n) = Θ(g(n))

$$\lim_{n \to \infty} \frac{\log n}{n} = ?$$

Undefined!

Recall L'Hôpital's rule?

Compare log n with n

$$\lim_{n\to\infty} \frac{f(n)}{g(n)}$$

0	f(n) = o(g(n))
∞	f(n) = ω(g(n))
k > 0	f(n) = Θ(g(n))

$$\lim_{n\to\infty} \frac{\log n}{n} = ?$$

Undefined!

Recall L'Hôpital's rule?

$$\lim_{n\to\infty} \frac{\log n}{n} = \lim_{n\to\infty} \frac{\frac{d}{dn}\log n}{\frac{d}{dn} n}$$

Compare log n with n

$$\lim_{n\to\infty} \frac{f(n)}{g(n)}$$

0	f(n) = o(g(n))
∞	f(n) = ω(g(n))
k > 0	f(n) = Θ(g(n))

$$\lim_{n\to\infty} \frac{\log n}{n} = ?$$

Undefined!

Recall L'Hôpital's rule?

$$\lim_{n\to\infty} \frac{\log n}{n} = \lim_{n\to\infty} \frac{\frac{d}{dn}\log n}{\frac{d}{dn} n}$$

= 1 / n

Compare log n with n

$$\lim_{n\to\infty} \frac{f(n)}{g(n)}$$

0	f(n) = o(g(n))
∞	f(n) = ω(g(n))
k > 0	f(n) = Θ(g(n))

$$\lim_{n\to\infty} \frac{\log n}{n} = ?$$

Undefined!

Recall L'Hôpital's rule?

$$\lim_{n\to\infty} \frac{\log n}{n} = \lim_{n\to\infty} \frac{\frac{d}{dn}\log n}{\frac{d}{dn} n}$$

= 1 / n

= 1

Compare log n with n

$$\lim_{n\to\infty} \frac{f(n)}{g(n)}$$

0	f(n) = o(g(n))
∞	f(n) = ω(g(n))
k > 0	f(n) = Θ(g(n))

$$\lim_{n\to\infty} \frac{\log n}{n} = ?$$

$$\lim_{n\to\infty} \frac{\log n}{n} = \lim_{n\to\infty} \frac{\frac{d}{dn}\log n}{\frac{d}{dn} n} = \lim_{n\to\infty} \frac{1}{n} = 0$$

Compare log n with n

$$\lim_{n\to\infty}\frac{f(n)}{g(n)}$$

0	**f(n) = o(g(n))**
∞	**f(n) = ω(g(n))**
k > 0	**f(n) = Θ(g(n))**

$$\lim_{n\to\infty}\frac{\log n}{n} = ?$$

log(n) = o(n)

$$\lim_{n\to\infty}\frac{\log n}{n} = \lim_{n\to\infty}\frac{\frac{d}{dn}\log n}{\frac{d}{dn}n} = \lim_{n\to\infty}\frac{1}{n} = 0$$

Compare log n with n

$$\lim_{n\to\infty} \frac{f(n)}{g(n)}$$

0	f(n) = o(g(n))
∞	f(n) = ω(g(n))
k > 0	f(n) = Θ(g(n))

$$\lim_{n\to\infty} \frac{\log n}{n} = ?$$

log(n) = O(n) but log(n) ≠ Ω(n)

$$\lim_{n\to\infty} \frac{\log n}{n} = \lim_{n\to\infty} \frac{\frac{d}{dn}\log n}{\frac{d}{dn} n} = \lim_{n\to\infty} \frac{1}{n} = 0$$

Compare $3n^2+5n+7$ with n^2

$$\lim_{n\to\infty} \frac{3n^2+5n+7}{n^2}$$

$$\lim_{n\to\infty} \frac{f(n)}{g(n)}$$

0	**f(n) = o(g(n))**
∞	**f(n) = ω(g(n))**
k > 0	**f(n) = Θ(g(n))**

Compare $3n^2+5n+7$ with n^2

$$\lim_{n\to\infty} \frac{f(n)}{g(n)}$$

0	f(n) = o(g(n))
∞	f(n) = ω(g(n))
k > 0	f(n) = Θ(g(n))

$$\lim_{n\to\infty} \frac{3n^2+5n+7}{n^2}$$

$$= \lim_{n\to\infty} 3 + \frac{5}{n} + \frac{7}{n^2}$$

= 3

Compare $3n^2+5n+7$ with n^2

$$\lim_{n\to\infty} \frac{f(n)}{g(n)}$$

0	f(n) = o(g(n))
∞	f(n) = ω(g(n))
k > 0	f(n) = Θ(g(n))

$$\lim_{n\to\infty} \frac{3n^2+5n+7}{n^2}$$

$$= \lim_{n\to\infty} 3 + \frac{5}{n} + \frac{7}{n^2}$$

$$= 3$$

$3n^2 + 5n + 7 = \Theta(n^2)$

Compare n^4 with 2^n

$$\lim_{n \to \infty} \frac{n^4}{2^n}$$

$$\lim_{n \to \infty} \frac{f(n)}{g(n)}$$

0	f(n) = o(g(n))
∞	f(n) = ω(g(n))
k > 0	f(n) = Θ(g(n))

Compare n^4 with 2^n

$$\lim_{n\to\infty} \frac{n^4}{2^n}$$

Undefined!

$$\lim_{n\to\infty} \frac{f(n)}{g(n)}$$

0	**f(n) = o(g(n))**
∞	**f(n) = ω(g(n))**
k > 0	**f(n) = Θ(g(n))**

Compare n^4 with 2^n

$$\lim_{n \to \infty} \frac{f(n)}{g(n)}$$

0	**f(n) = o(g(n))**
∞	**f(n) = ω(g(n))**
k > 0	**f(n) = Θ(g(n))**

$$\lim_{n \to \infty} \frac{n^4}{2^n}$$

$$= \lim_{n \to \infty} \frac{\frac{d}{dn} n^4}{\frac{d}{dn} 2^n}$$

Compare n^4 with 2^n

$$\lim_{n\to\infty} \frac{f(n)}{g(n)}$$

0	f(n) = o(g(n))
∞	f(n) = ω(g(n))
k > 0	f(n) = Θ(g(n))

$$\lim_{n\to\infty} \frac{n^4}{2^n}$$

$$= \lim_{n\to\infty} \frac{\frac{d}{dn} n^4}{\frac{d}{dn} 2^n}$$

Compare n^4 with 2^n

$$\lim_{n\to\infty} \frac{f(n)}{g(n)}$$

0	f(n) = o(g(n))
∞	f(n) = ω(g(n))
k > 0	f(n) = Θ(g(n))

$$\lim_{n\to\infty} \frac{n^4}{2^n}$$

$$= \lim_{n\to\infty} \frac{\frac{d}{dn} n^4}{\frac{d}{dn} 2^n}$$

$$2 = e^{\ln 2}$$

Compare n^4 with 2^n

$$\lim_{n\to\infty} \frac{f(n)}{g(n)}$$

0	**f(n) = o(g(n))**
∞	**f(n) = ω(g(n))**
k > 0	**f(n) = Θ(g(n))**

$$\lim_{n\to\infty} \frac{n^4}{2^n}$$

$$= \lim_{n\to\infty} \frac{\frac{d}{dn} n^4}{\frac{d}{dn} 2^n}$$

$$2 = e^{\ln 2}$$

$$2^n = \left(e^{\ln 2}\right)^n$$

Compare n^4 with 2^n

$$\lim_{n\to\infty} \frac{f(n)}{g(n)}$$

0	**f(n) = o(g(n))**
∞	**f(n) = ω(g(n))**
k > 0	**f(n) = Θ(g(n))**

$$\lim_{n\to\infty} \frac{n^4}{2^n}$$

$$= \lim_{n\to\infty} \frac{\frac{d}{dn} n^4}{\frac{d}{dn} 2^n}$$

$$2 = e^{\ln 2}$$

$$2^n = \left(e^{\ln 2}\right)^n = e^{(\ln 2)n}$$

Compare n^4 with 2^n

$$\lim_{n\to\infty} \frac{f(n)}{g(n)}$$

0	f(n) = o(g(n))
∞	f(n) = ω(g(n))
k > 0	f(n) = Θ(g(n))

$$\lim_{n\to\infty} \frac{n^4}{2^n}$$

$$= \lim_{n\to\infty} \frac{\frac{d}{dn} n^4}{\frac{d}{dn} 2^n}$$

$$2 = e^{\ln 2}$$

$$2^n = \left(e^{\ln 2}\right)^n = e^{(\ln 2)n}$$

$$\frac{d}{dn} e^{(\ln 2)n} = \ln 2 \; e^{(\ln 2)n}$$

Compare n^4 with 2^n

$$\lim_{n\to\infty} \frac{f(n)}{g(n)}$$

0	f(n) = o(g(n))
∞	f(n) = ω(g(n))
k > 0	f(n) = Θ(g(n))

$$\lim_{n\to\infty} \frac{n^4}{2^n}$$

$$= \lim_{n\to\infty} \frac{\frac{d}{dn} n^4}{\frac{d}{dn} 2^n}$$

$$\frac{d}{dn} e^{(\ln 2)n} = \ln 2 \; e^{(\ln 2)n}$$

Compare n^4 with 2^n

$$\lim_{n\to\infty} \frac{f(n)}{g(n)}$$

0	**f(n)**	**= o(g(n))**
∞	**f(n)**	**= ω(g(n))**
k > 0	**f(n)**	**= Θ(g(n))**

$$\lim_{n\to\infty} \frac{n^4}{2^n}$$

$$= \lim_{n\to\infty} \frac{\frac{d}{dn} n^4}{\frac{d}{dn} 2^n}$$

$$\frac{d}{dn} e^{(\ln 2)n} = \ln 2 \; e^{(\ln 2)n}$$

$$= \ln 2 \; 2^n$$

Compare n^4 with 2^n

$$\lim_{n\to\infty}\frac{f(n)}{g(n)}$$

0	f(n) = o(g(n))
∞	f(n) = ω(g(n))
k > 0	f(n) = Θ(g(n))

$$\lim_{n\to\infty}\frac{n^4}{2^n}$$

$$=\lim_{n\to\infty}\frac{\frac{d}{dn}n^4}{\frac{d}{dn}2^n}$$

$$=\ln 2\ 2^n$$

Compare n^4 with 2^n

$$\lim_{n\to\infty} \frac{f(n)}{g(n)}$$

0	f(n) = o(g(n))
∞	f(n) = ω(g(n))
k > 0	f(n) = Θ(g(n))

$$\lim_{n\to\infty} \frac{n^4}{2^n}$$

$$= \lim_{n\to\infty} \frac{\frac{d}{dn} n^4}{\frac{d}{dn} 2^n} = \lim_{n\to\infty} \frac{4n^3}{\ln 2 \; 2^n}$$

Compare n^4 with 2^n

$$\lim_{n\to\infty}\frac{f(n)}{g(n)}$$

0	**f(n) = o(g(n))**
∞	**f(n) = ω(g(n))**
k > 0	**f(n) = Θ(g(n))**

$$\lim_{n\to\infty}\frac{n^4}{2^n}$$

$$= \lim_{n\to\infty}\frac{\frac{d}{dn}n^4}{\frac{d}{dn}2^n} = \lim_{n\to\infty}\frac{4n^3}{\ln 2\ 2^n}$$

Compare n^4 with 2^n

$$\lim_{n\to\infty} \frac{f(n)}{g(n)}$$

0	**f(n) = o(g(n))**
∞	**f(n) = ω(g(n))**
k > 0	**f(n) = Θ(g(n))**

$$\lim_{n\to\infty} \frac{n^4}{2^n}$$

Keep applying L'Hôpital's rule

$$= \lim_{n\to\infty} \frac{\frac{d}{dn} n^4}{\frac{d}{dn} 2^n} = \lim_{n\to\infty} \frac{4n^3}{\ln 2 \; 2^n}$$

Compare n^4 with 2^n

$$\lim_{n\to\infty} \frac{f(n)}{g(n)}$$

0	**f(n) = o(g(n))**
∞	**f(n) = ω(g(n))**
k > 0	**f(n) = Θ(g(n))**

$$\lim_{n\to\infty} \frac{n^4}{2^n}$$

Keep applying L'Hôpital's rule

$$= \lim_{n\to\infty} \frac{\frac{d}{dn} n^4}{\frac{d}{dn} 2^n} = \lim_{n\to\infty} \frac{4n^3}{\ln 2 \ 2^n} = \lim_{n\to\infty} \frac{12n^2}{(\ln 2)^2 \ 2^n}$$

Compare n^4 with 2^n

$$\lim_{n\to\infty} \frac{f(n)}{g(n)}$$

0	f(n) = o(g(n))
∞	f(n) = ω(g(n))
k > 0	f(n) = Θ(g(n))

$$\lim_{n\to\infty} \frac{n^4}{2^n}$$

Keep applying L'Hôpital's rule

$$= \lim_{n\to\infty} \frac{\frac{d}{dn} n^4}{\frac{d}{dn} 2^n} = \lim_{n\to\infty} \frac{4n^3}{\ln 2 \ 2^n} = \lim_{n\to\infty} \frac{12n^2}{(\ln 2)^2 \ 2^n}$$

$$= \lim_{n\to\infty} \frac{24n}{(\ln 2)^3 \ 2^n}$$

Compare n^4 with 2^n

$$\lim_{n\to\infty}\frac{f(n)}{g(n)}$$

0	f(n) = o(g(n))
∞	f(n) = ω(g(n))
k > 0	f(n) = Θ(g(n))

$$\lim_{n\to\infty}\frac{n^4}{2^n}$$

Keep applying L'Hôpital's rule

$$= \lim_{n\to\infty}\frac{\frac{d}{dn}n^4}{\frac{d}{dn}2^n} = \lim_{n\to\infty}\frac{4n^3}{\ln 2\ 2^n} = \lim_{n\to\infty}\frac{12n^2}{(\ln 2)^2\ 2^n}$$

$$= \lim_{n\to\infty}\frac{24n}{(\ln 2)^3\ 2^n} = \lim_{n\to\infty}\frac{24}{(\ln 2)^4\ 2^n} = 0$$

Compare n^4 with 2^n

$$\lim_{n\to\infty} \frac{f(n)}{g(n)}$$

0	**f(n) = o(g(n))**
∞	**f(n) = ω(g(n))**
k > 0	**f(n) = Θ(g(n))**

$$\lim_{n\to\infty} \frac{n^4}{2^n}$$

$$= \lim_{n\to\infty} \frac{\frac{d}{dn} n^4}{\frac{d}{dn} 2^n} = \lim_{n\to\infty} \frac{4n^3}{\ln 2 \; 2^n} = \lim_{n\to\infty} \frac{12n^2}{(\ln 2)^2 \; 2^n}$$

$$= \lim_{n\to\infty} \frac{24n}{(\ln 2)^3 \; 2^n} = \lim_{n\to\infty} \frac{24}{(\ln 2)^4 \; 2^n} = \boxed{0}$$

Compare n^4 with 2^n

$$\lim_{n\to\infty} \frac{f(n)}{g(n)}$$

0	f(n) = o(g(n))
∞	f(n) = ω(g(n))
k > 0	f(n) = Θ(g(n))

$$\lim_{n\to\infty} \frac{n^4}{2^n} = 0 \Rightarrow$$ **$n^4 = o(2^n)$**

$$= \lim_{n\to\infty} \frac{\frac{d}{dn} n^4}{\frac{d}{dn} 2^n} = \lim_{n\to\infty} \frac{4n^3}{\ln 2 \; 2^n} = \lim_{n\to\infty} \frac{12n^2}{(\ln 2)^2 \; 2^n}$$

$$= \lim_{n\to\infty} \frac{24n}{(\ln 2)^3 \; 2^n} = \lim_{n\to\infty} \frac{24}{(\ln 2)^4 \; 2^n} = 0$$

Compare n^4 with 2^n

$$\lim_{n\to\infty} \frac{f(n)}{g(n)}$$

0	f(n) = o(g(n))
∞	f(n) = ω(g(n))
k > 0	f(n) = Θ(g(n))

$$\lim_{n\to\infty} \frac{n^4}{2^n} = 0 \Rightarrow$$

$n^4 = O(2^n)$ but $n^4 \neq \Omega(2^n)$

$$= \lim_{n\to\infty} \frac{\frac{d}{dn} n^4}{\frac{d}{dn} 2^n} = \lim_{n\to\infty} \frac{4n^3}{\ln 2 \; 2^n} = \lim_{n\to\infty} \frac{12n^2}{(\ln 2)^2 \; 2^n}$$

$$= \lim_{n\to\infty} \frac{24n}{(\ln 2)^3 \; 2^n} = \lim_{n\to\infty} \frac{24}{(\ln 2)^4 \; 2^n} = 0$$

More generally

Can show inductively that for any *real-valued* a > 1, b ≥ 0,

$$n^b = O(a^n) \quad \textbf{but} \quad n^b \neq \Omega(a^n)$$

"Exponentials grow **faster** than polynomials"

Interlude

Given some algorithms, which is fastest?

Algorithm	Time
A1	$\Theta(n^2)$
A2	$\Theta(n \log n)$
A3	$\Omega(n^2)$
A4	$\Theta(n^3)$
A5	$\Theta(n \log n)$

Given some algorithms, which is fastest?

Algorithm	Time
	$\Theta(n^2)$
	$\Theta(n \log n)$
	$\Omega(n^2)$
	$\Theta(n^3)$
	$\Theta(n \log n)$

These are tight bounds, so they can easily be compared against each other.

Either n log n time is better than the other two times

Given some algorithms, which is fastest?

Algorithm	Time
A1	$\Theta(n^2)$
A2	$\Theta(n \log n)$
A3	$\Omega(n^2)$
A4	$\Theta(n^3)$
A5	$\Theta(n \log n)$

Given some algorithms, which is fastest?

Algorithm	Time
A1	$\Theta(n^2)$
A2	$\Theta(n \log n)$
A3	$\Omega(n^2)$
A4	$\Theta(n^3)$
A5	$\Theta(n \log n)$

Given some algorithms, which is fastest?

Algorithm	Time
	$\Theta(n^2)$
	$\Theta(n \log n)$
A3	$\Omega(n^2)$
A4	$\Theta(n^3)$
A5	$\Theta(n \log n)$

Not a tight bound, but this algorithm takes *at least* n^2 time.

Thus it is slower than A2 or A5

Given some algorithms, which is fastest?

Algorithm	Time
A1	$\Theta(n^2)$
A2	$\Theta(n \log n)$
A5	$\Theta(n \log n)$

How do we decide between A2 and A5?

Given some algorithms, which is fastest?

Algorithm	Time
A1	$\Theta(n^2)$
A2	$\Theta(n \log n)$
A5	$\Theta(n \log n)$

- **Constant factors**
- **Empirical behavior**
- **Properties other than speed**

Given some algorithms, which is fastest?

Algorithm	Time
A1	$\Theta(n^2)$
A2	$\Theta(n \log n)$
A5	$\Theta(n \log n)$

Could I sell you an $O(n^2)$ algorithm? Would you use it instead of A2 or A5?

Given an algorithm, which is best? A2 or A5 so far

Algorithm	Time
A1	$\Theta(n^2)$
A2	$\Theta(n \log n)$
A5	$\Theta(n \log n)$

- **An $O(n^2)$ algorithm could be faster than A2 or A5**
- **But it could also be slower!**
- **We can't tell without a lower bound**

And Now For Something Completely Different

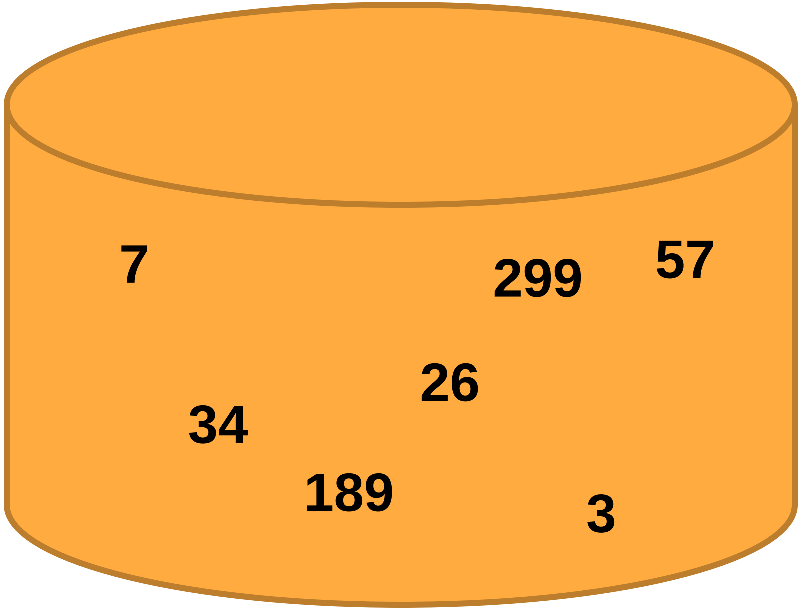
- We’ve been focused on how to express, compare running times.

- Now we’re going to put that knowledge into practice!

- We’ll start with basic data structures.

Collection Abstract Data Types

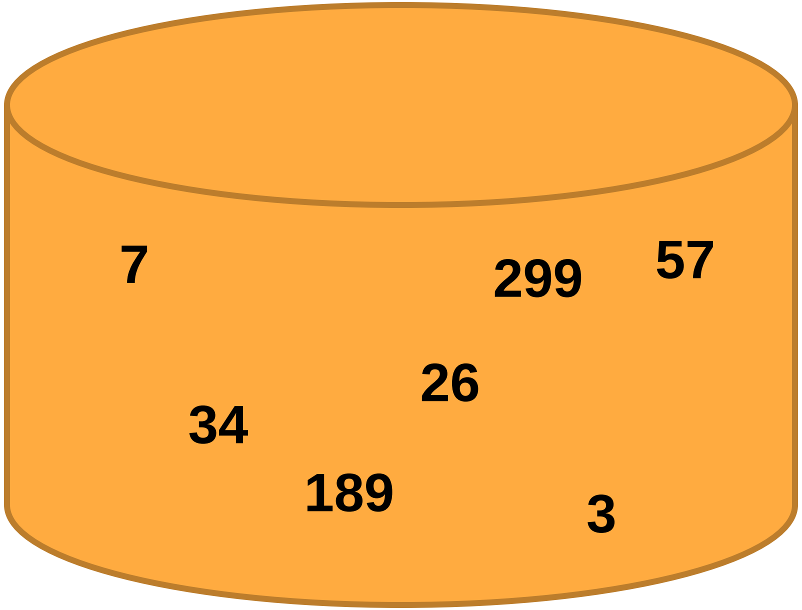
- A collection is just a bunch of objects (of some common type)



Collection Abstract Data Types

- **A collection is just a bunch of objects (of some common type)**

- Numbers
- Strings
- Records
- …



Collection Abstract Data Types

- A collection is just a bunch of objects (of some common type)
- Each object has a key, and maybe some other attached data

Name: J. Random Hacker
Student ID: 247247
Year: Sophomore
Home Town: Kalamazoo

Collection Abstract Data Types

- A collection is just a bunch of objects (of some common type)
- Each object has a key, and maybe some other attached data
- (We usually focus on the keys and ignore the rest)

Things We Might Do With a Collection

- Enumerate the keys of all objects
- Add an object
- Remove an object
- Find an object by key

Things We Might Do With a Collection

- Enumerate the keys of all objects
- Add an object
- Remove an object
- Find an object by key
- Locate largest/smallest key

Things We Might Do With a Collection

- Enumerate the keys of all objects
- Add an object
- Remove an object
- Find an object by key
- Locate largest/smallest key

Assumes keys are ordered! (More later)

ADTs Have Methods – for Example:

- **Enumerate()**
- **Add(key)**
- **Remove(key)**
- **Find(key)**
- **Max(), Min()**

ADTs Have Methods – for Example:

- Enumerate()
- Add(key)
- Remove(key)
- Find(key)
- Max(), Min()

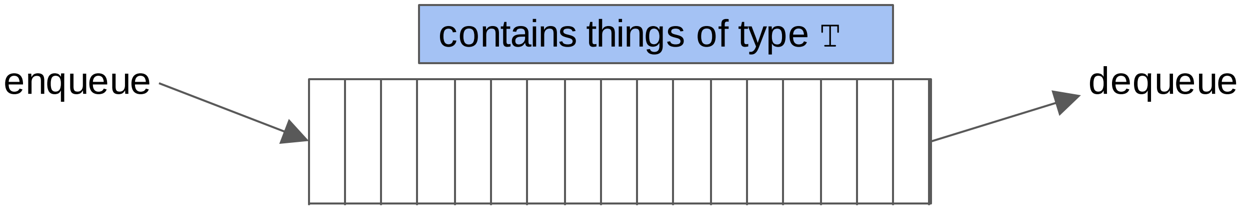
We know how these methods act on the collection, but not how they are implemented

Some Collections are Structured

- Objects may be logically arranged inside a collection.

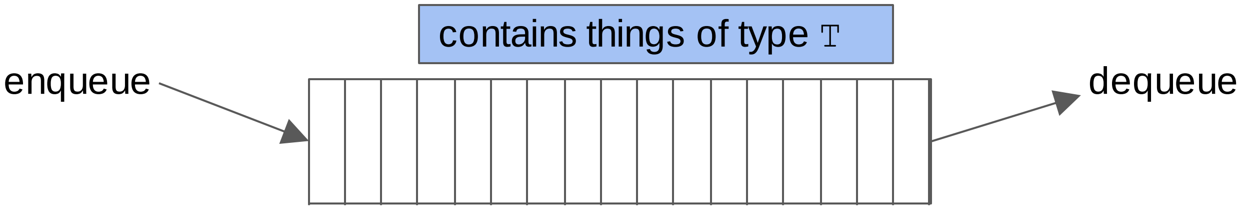
Example: Queue (Java notation)

- `Queue<T>` : 2 basic operations



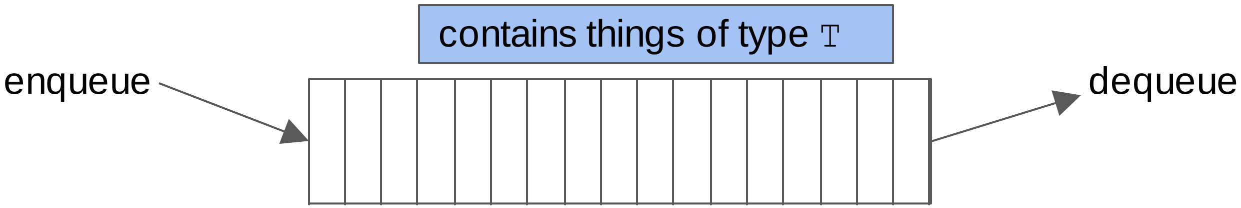
Example: Queue

- `Queue<T>` : 2 basic operations
  - `void enqueue(T thing)`
    - Adds `thing` to the end of the queue



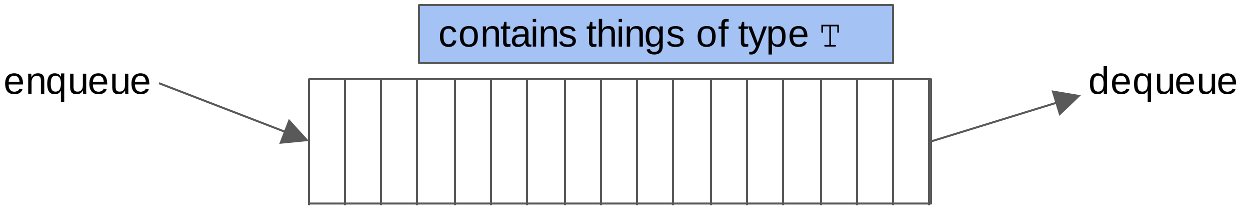
Example: Queue

- `Queue<T>` : 2 basic operations
  - `void enqueue(T thing)`
    - Adds `thing` to the end of the queue
  - `T dequeue()`
    - Removes and returns the thing at the beginning of the queue
      - Fails if the queue is empty



Example: Queue

- `Queue<T>` : 2 basic operations
  - `void enqueue(T thing)`
    - Adds `thing` to the end of the queue
  - `T dequeue()`
    - Removes and returns the thing at the beginning of the queue
      - Fails if the queue is empty
  - `boolean isEmpty()`
    - Returns whether the queue is empty

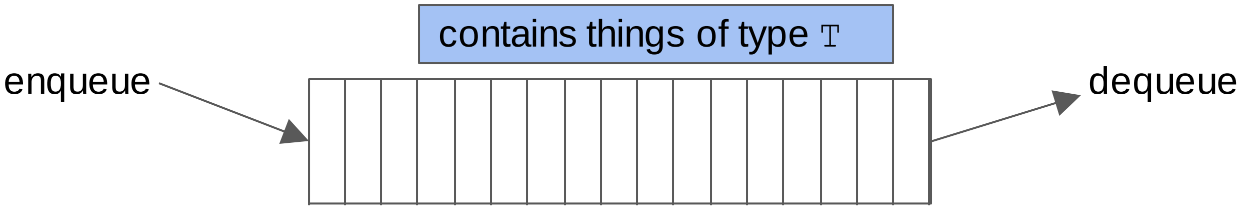


Example: Queue

- `Queue<T>` : 2 basic operations
  - `void enqueue(T thing)`
    - Adds `thing` to the end of the queue
  - `T dequeue()`
    - Removes and retur… …g at the beginning of the q…
      - Fails if the queue is empty
  - `boolean isEmpty()`
    - Returns whether the queue is empty

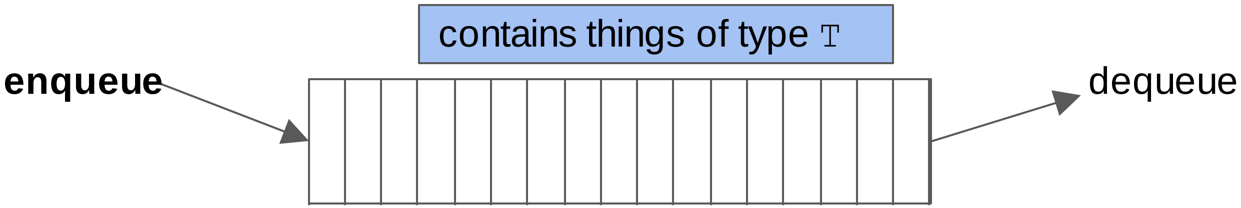
What does it mean to "fail"? Two choices:

- return null
- throw an Exception NoSuchElement



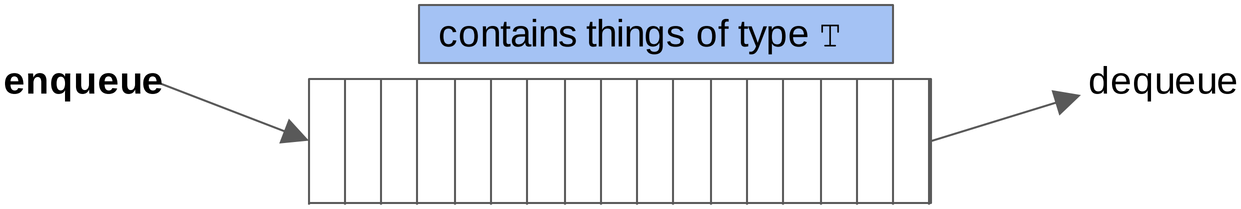
Example: Queue

- `Queue<T>`
  - Key characteristic: FIFO order
    - First In, First Out



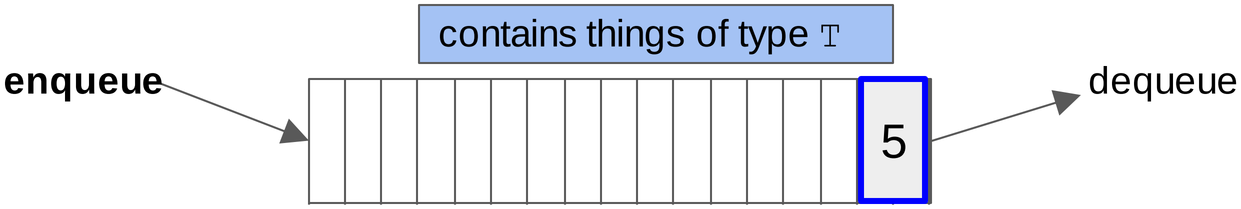
Example: Queue

- `Queue<T>`
  - Key characteristic: FIFO order
    - First In, First Out
  - Example: add, then remove objects holding 5, 3, and 8; note FIFO order of removal



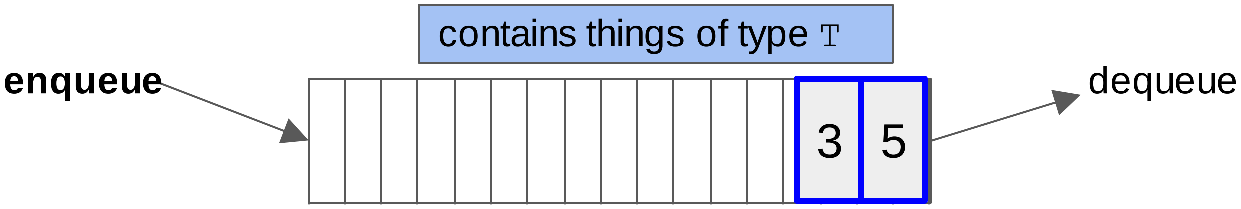
Example: Queue

- `Queue<T>`
  - Key characteristic: FIFO order
    - First In, First Out
  - Example:
    - Add: enqueue(5)



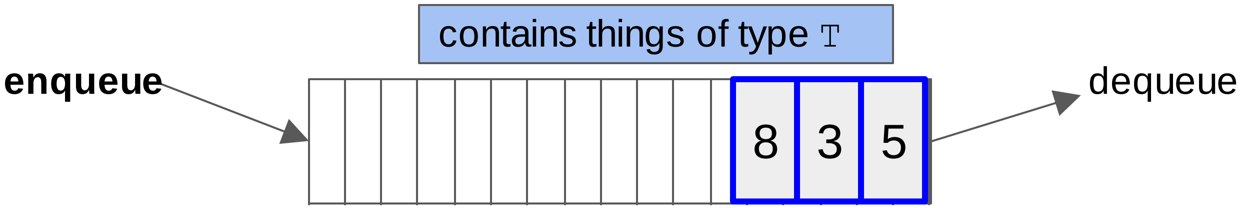
Example: Queue

- Queue<T>
  - Key characteristic: FIFO order
    - First In, First Out
  - Example:
    - Add: enqueue(5), enqueue(3)



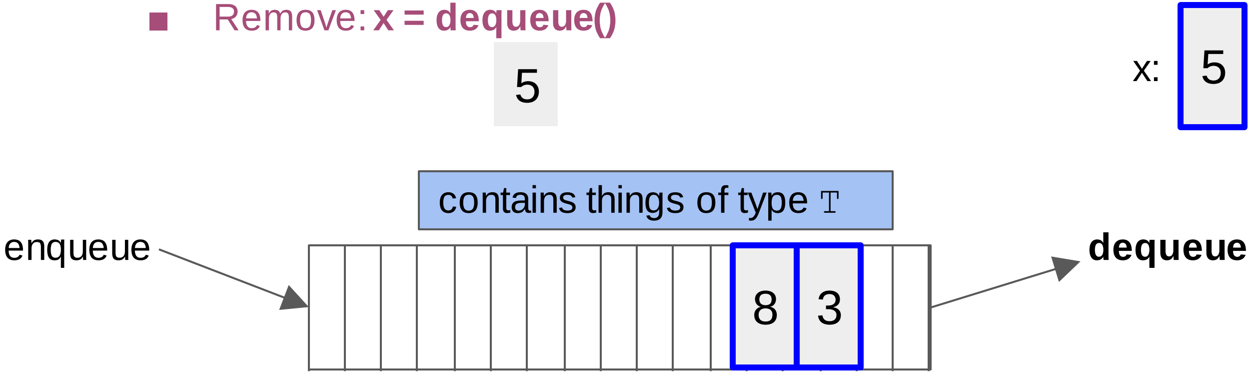
Example: Queue

- `Queue<T>`
  - Key characteristic: FIFO order
    - First In, First Out
  - Example:
    - Add: enqueue(5), enqueue(3), enqueue(8)



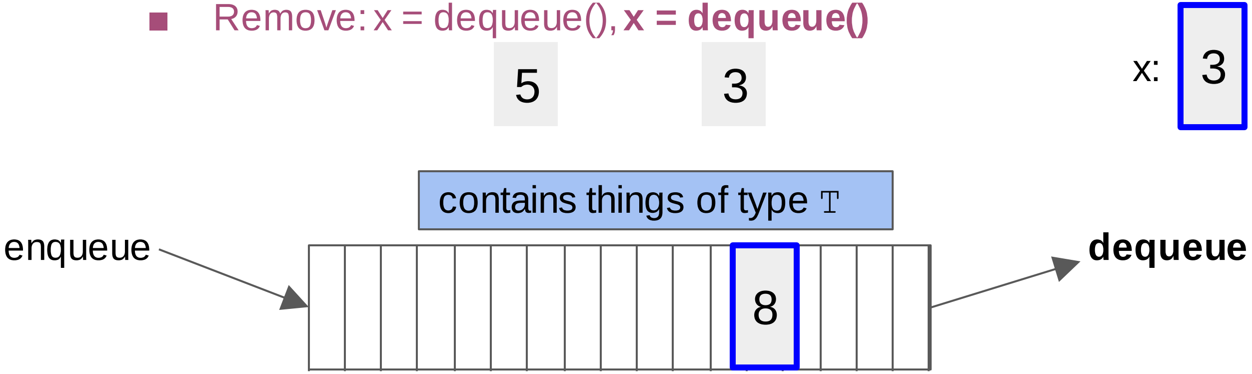
Example: Queue

- `Queue<T>`
  - Key characteristic: FIFO order
    - First In, First Out
  - Example:
    - Add: enqueue(5), enqueue(3), enqueue(8)
    - Remove: **x = dequeue()**



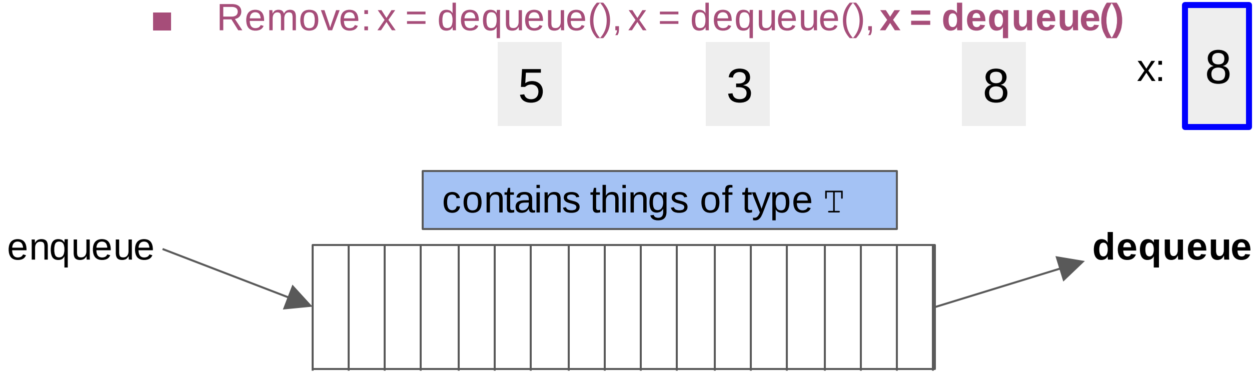
Example: Queue

- Queue<T>
  - Key characteristic: FIFO order
    - First In, First Out
  - Example:
    - Add: enqueue(5), enqueue(3), enqueue(8)
    - Remove: x = dequeue(), **x = dequeue()**



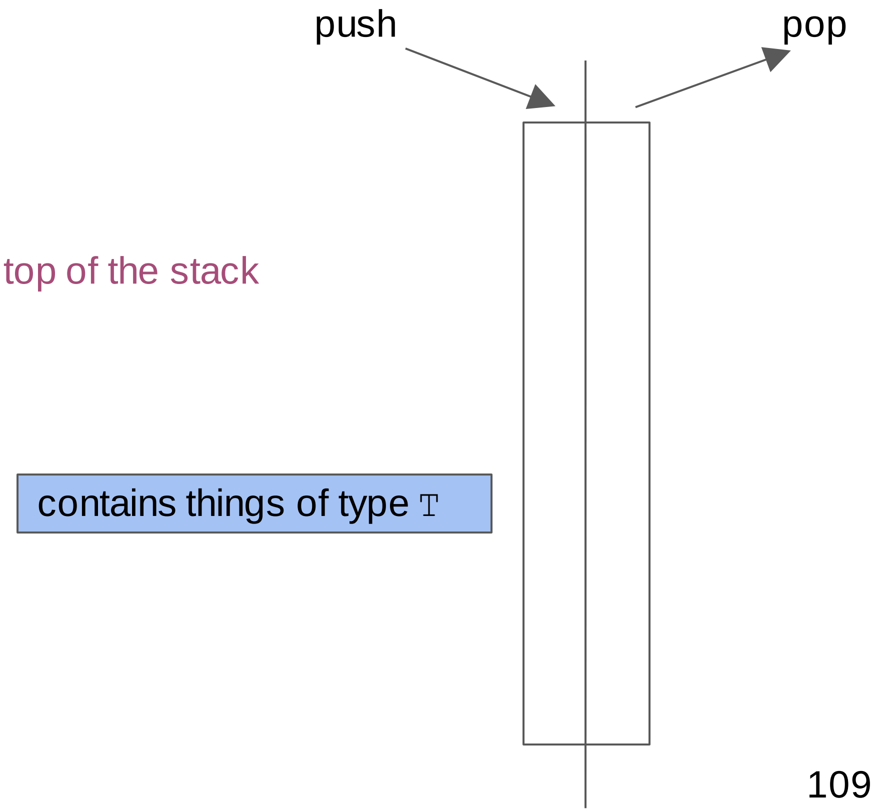
Example: Queue

- `Queue<T>`
  - Key characteristic: FIFO order
    - First In, First Out
  - Example:
    - Add: enqueue(5), enqueue(3), enqueue(8)
    - Remove: x = dequeue(), x = dequeue(), **x = dequeue()**



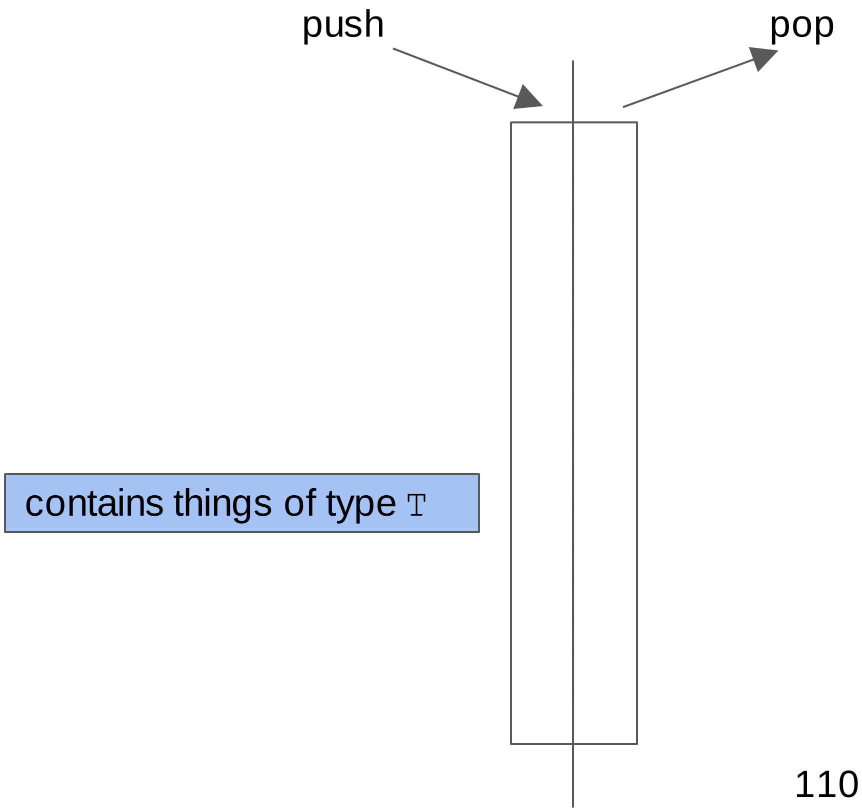
Example: Stack (Java notation)

- Stack<T> : 2 basic operations
  - void push(T thing)
    - Adds thing to the top of the stack
  - T pop()
    - Removes and returns the thing at the top of the stack
      - Fails if the stack is empty
  - Key characteristic: LIFO order
    - Last In, First Out



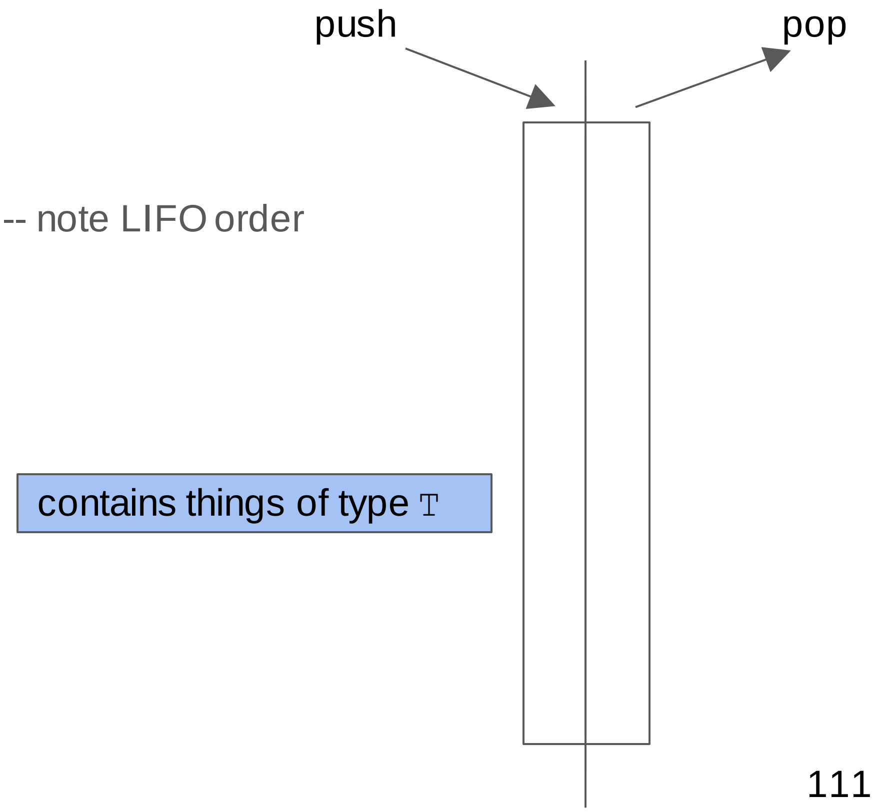
Example: Stack

- `Stack<T>` : 2 basic operations
  - Key characteristic: LIFO order
    - Last In, First Out



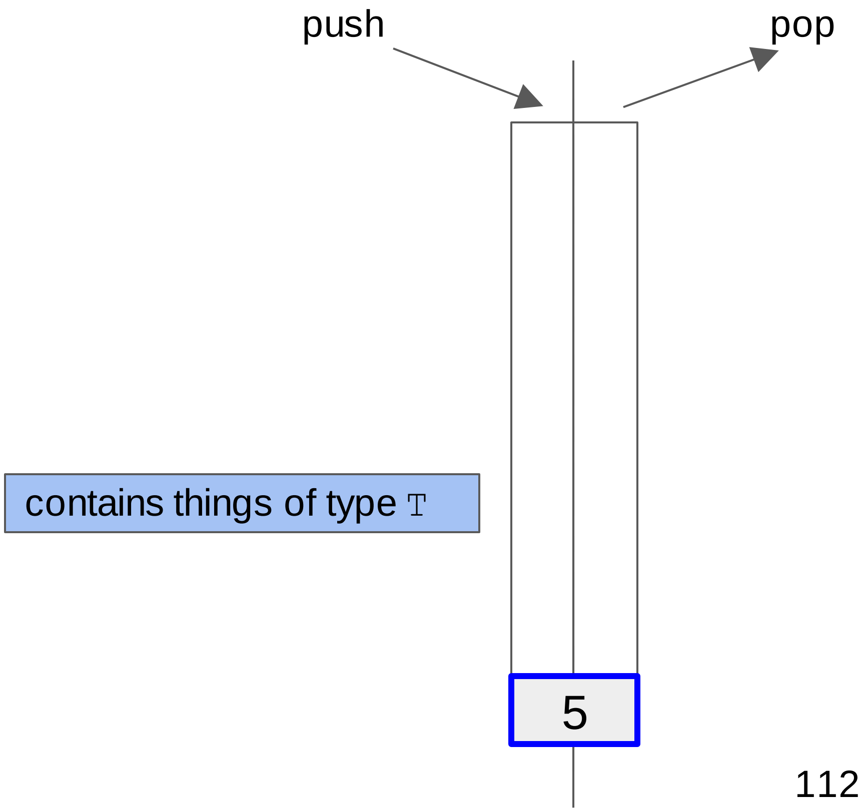
Example: Stack

- `Stack<T>` : 2 basic operations
  - Key characteristic: LIFO order
    - Last In, First Out
  - Example: add 5, 3, 8 to stack, then remove -- note LIFO order
    - Add:



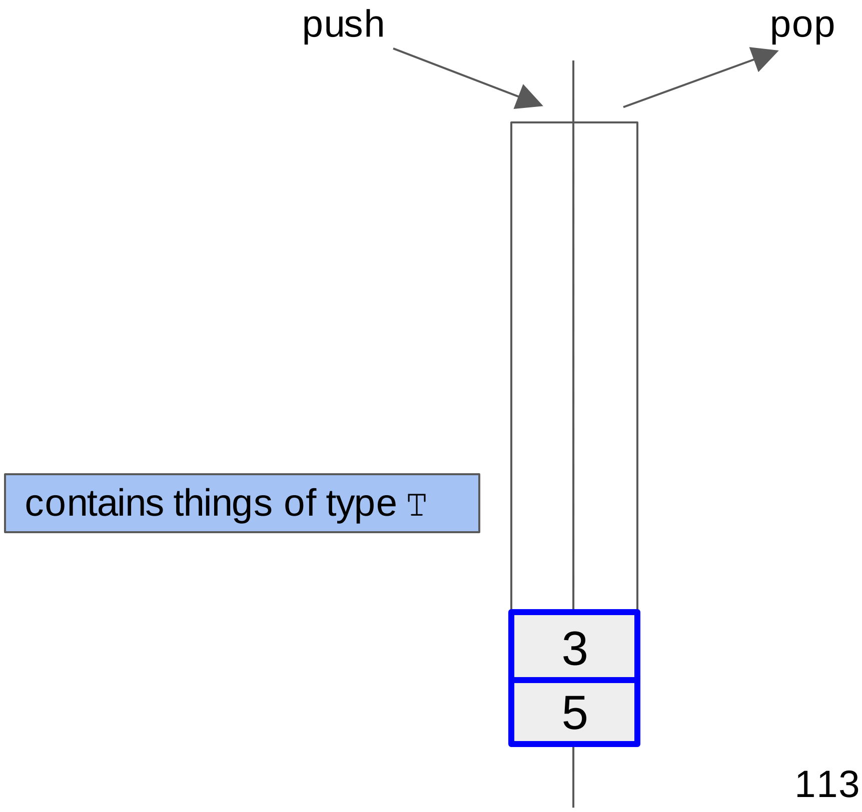
Example: Stack

- `Stack<T>` : 2 basic operations
  - Key characteristic: LIFO order
    - Last In, First Out
  - Example
    - Add: push(5)



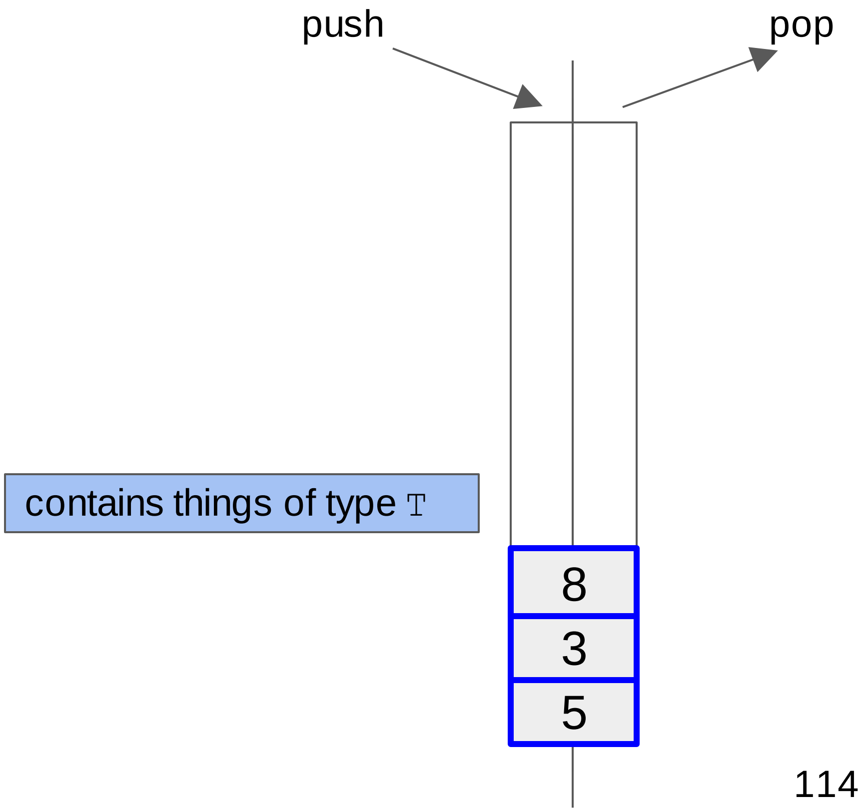
Example: Stack

- `Stack<T>` : 2 basic operations
  - Key characteristic: LIFO order
    - Last In, First Out
  - Example
    - Add: push(5), push(3)



Example: Stack

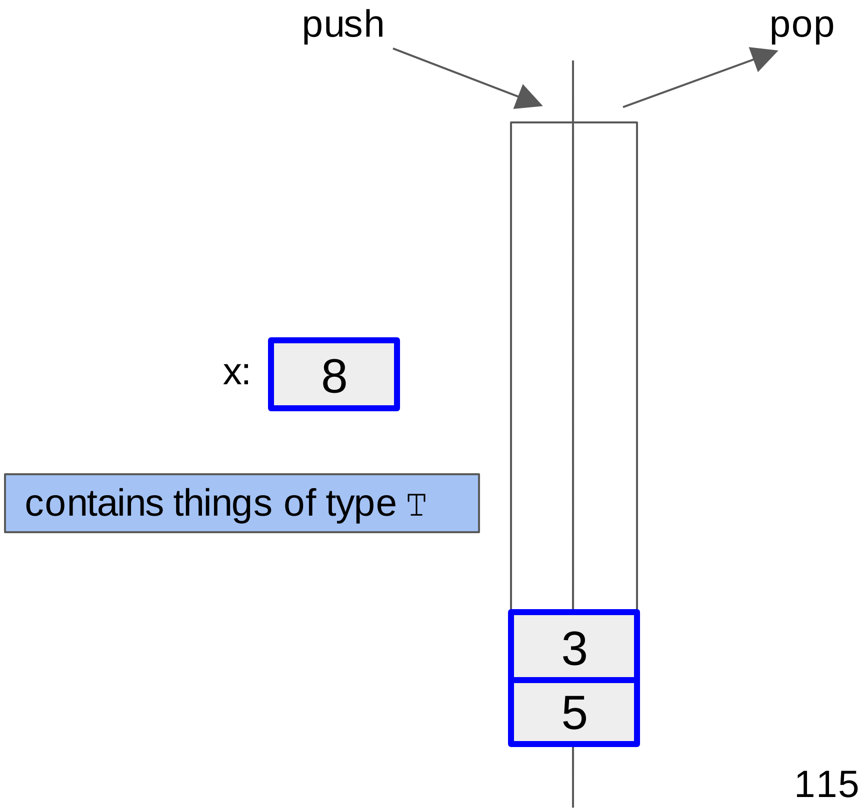
- `Stack<T>` : 2 basic operations
  - Key characteristic: LIFO order
    - Last In, First Out
  - Example
    - Add: push(5), push(3), push(8)



Example: Stack

- `Stack<T>` : 2 basic operations
  - Key characteristic: LIFO order
    - Last In, First Out
  - Example
    - Add: push(5), push(3), push(8)
    - Remove: **x=pop()**

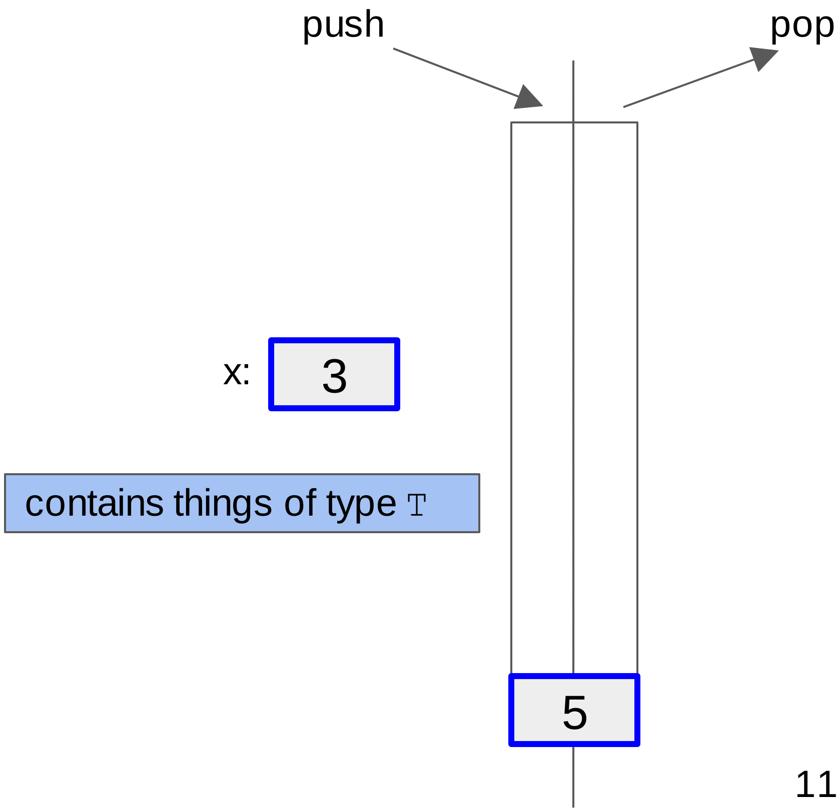
8



Example: Stack

- `Stack<T>` : 2 basic operations
    - Key characteristic: LIFO order
        - Last In, First Out
    - Example
        - Add: push(5), push(3), push(8)
        - Remove: x=pop(), **x=pop()**

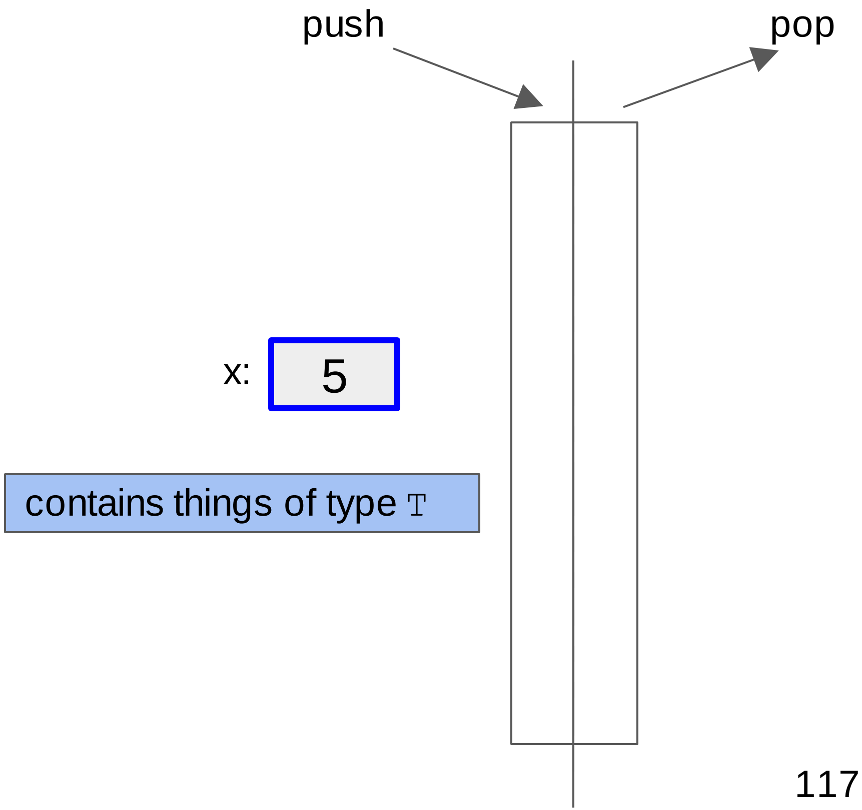
8 3



Example: Stack

- `Stack<T>` : 2 basic operations
  - Key characteristic: LIFO order
    - Last In, First Out
  - Example
    - Add: push(5), push(3), push(8)
    - Remove: x=pop(), x=pop(), **x=pop()**

8 3 5

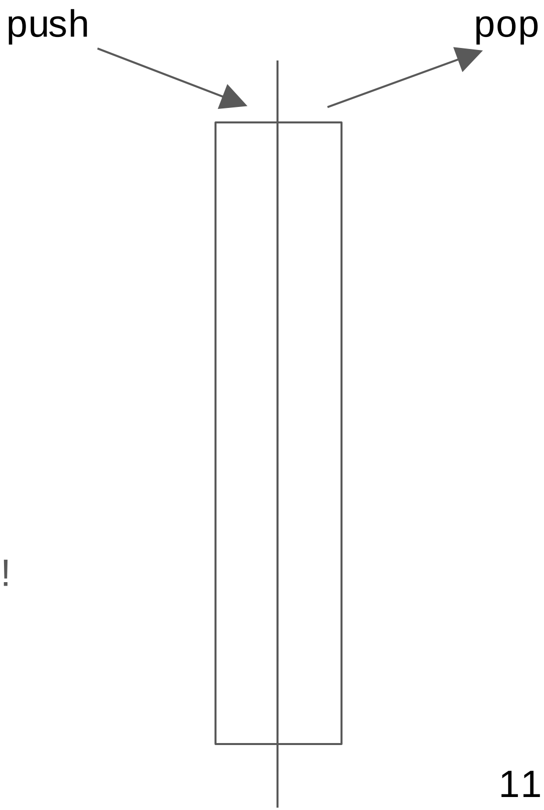


Example: Stack

- `Stack<T>` : 2 basic operations
  - Key characteristic: LIFO order
    - Last In, First Out
  - Example
    - Add: push(5), push(3), push(8)
    - Remove: x=pop(), x=pop(), **x=pop()**

8 3 5

  - Note: elements come out in reverse order compared to Queue!
    - LIFO vs. FIFO



An ADT can be implemented in different ways.

Example: Queue Implementations

- **Our picture of a Queue suggests an array**

Example: Queue Implementations

- Our picture of a Queue suggests an array

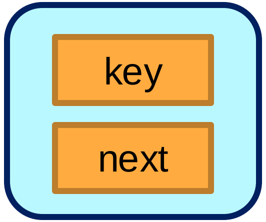
- Idea: maintain two pointers – “head” and “tail”

Example: Queue Implementations

- Our picture of a Queue suggests an array
- Idea: maintain two pointers – “head” and “tail”
- Enqueue items at the tail
- Dequeue items at the head
- (Maintains FIFO ordering)

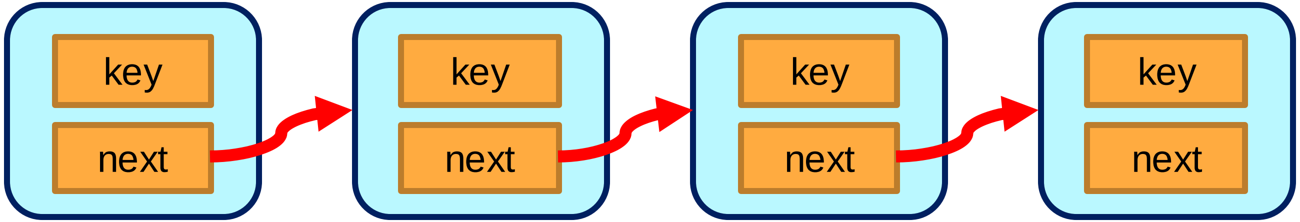
Alternative Structure: Linked List

- We could implement the same behavior using a linked list
- A list consists of nodes, each of which holds a key (object).



Alternative Structure: Linked List

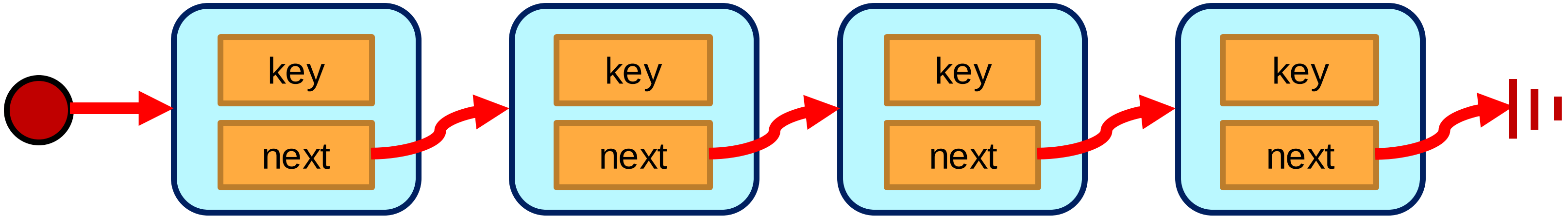
- We could implement the same behavior using a linked list
- A list consists of nodes, each of which holds a key (object).



- Each node's *next pointer* points to its successor.

Alternative Structure: Linked List

- A basic linked list has a head pointer to its first node.
- The last node's next pointer is null.



(There are fancier lists, e.g. *doubly-linked* with next *and* previous pointers, but we'll focus on this basic list for now.)

Implementing a Queue With a List

- **How can we map the basic Queue operations onto a linked list?**

- **Need enqueue, dequeue**

Implementing a Queue With a List (One Way)

- Enqueue items at the end of the list
- Dequeue from the beginning of the list
- **FIFO order is preserved!**

Performance Implications

- In an array-based queue, cost of enqueue, dequeue is ???

Performance Implications

- **In an array-based queue, cost of enqueue, dequeue is $\Theta(1)$**
- **[read/write one element and bump a pointer]**

Performance Implications

- **In an array-based queue, cost of enqueue, dequeue is Θ(1)**
- **[read/write one element and bump a pointer]**

- **But for our list-based queue, enqueue must find the tail to add a new object.**
- **Moving from head to tail requires following Θ(n) pointers in an n-element list.**

Performance of an ADT is sensitive to the data structure used to implement it.

What's Next?

- We'll look at an ADT for which neither arrays nor lists provide satisfactory performance for all operations.
- We'll see an entirely new data structure to implement it.
- We'll reason about the performance of this structure.