Applications of Stacks

- The runtime stack in memory
- Converting a recursive algorithm to an iterative one
  - use a stack to emulate the runtime stack
- Making sure that delimiters (parens, brackets, etc.) are balanced:
  - push open (i.e., left) delimiters onto a stack
  - when you encounter a close (i.e., right) delimiter, pop an item off the stack and see if it matches
  - example: $5 \times [3 + \{(5 + 16 - 2)\}$
Applications of Stacks

• The runtime stack in memory
• Converting a recursive algorithm to an iterative one
  • use a stack to emulate the runtime stack
• Making sure that delimiters (parens, brackets, etc.) are balanced:
  • push open (i.e., left) delimiters onto a stack
  • when you encounter a close (i.e., right) delimiter, pop an item off the stack and see if it matches
  • example: \( 5 \times [3 + \{5 + 16 - 2\}] \)

\[
\begin{align*}
push \, [ & \quad push \, \{ \quad push \, ( \\
[ & \quad \{ \\
[ & \quad \\
[ & \\
\end{align*}
\]

\[
\begin{align*}
push \, [ & \quad push \, \{ \quad push \, ( \\
[ & \quad \{ \\
[ & \quad \\
[ & \\
\end{align*}
\]
Applications of Stacks

- The runtime stack in memory
- Converting a recursive algorithm to an iterative one
  - use a stack to emulate the runtime stack
- Making sure that delimiters (parens, brackets, etc.) are balanced:
  - push open (i.e., left) delimiters onto a stack
  - when you encounter a close (i.e., right) delimiter, pop an item off the stack and see if it matches
  - example: $5 \times [3 + \{(5 + 16 - 2)\}$

```
push [       push {       push ( (       ), so pop.
[          ]          ]          )
```

```
push [       push {       push ( (       ), so pop.
[          ]          ]          )
```

get (, which
matches
Applications of Stacks

- The runtime stack in memory
- Converting a recursive algorithm to an iterative one
  - use a stack to emulate the runtime stack
- Making sure that delimiters (parens, brackets, etc.) are balanced:
  - push open (i.e., left) delimiters onto a stack
  - when you encounter a close (i.e., right) delimiter, pop an item off the stack and see if it matches
- example: \( 5 \times [3 + \{(5 + 16 - 2)\}] \)

```
push [  
push {  
push (  
push )  
pop ( ,  
which matches  
pop. 
```

```
push [  
push {  
push (  
push )  
pop ( ,  
which doesn't match  
```