Stack

Data structures: Stack

- **Stack** is a data structure in which:
    - Items can be inserted only from one end.
    - Items can be taken only from the **same** end.

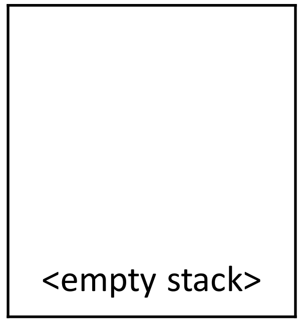
- The last inserted item is the first item to be taken.
    - Last Input First Output [LIFO].

- Example:
    - Stack of plates.

Data structures: Stack Operations

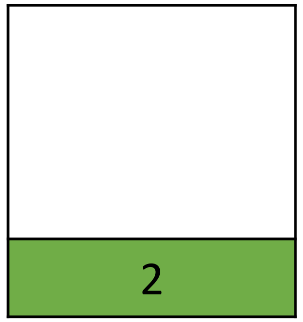
- **push:** Inserts item to the top of the stack. Time complexity is O(1).
- **pop:** Removes items from the top of stack. Time complexity is O(1).
- **top:** Returns top element of the stack. Time complexity is O(1).
- **empty:** Returns “true”, if stack is empty. Time complexity is O(1).
- **size:** Returns size of stack. Time complexity is O(1).

Data structures: Stack Example

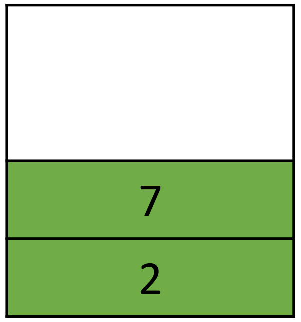


Data structures: Stack Example

push 2



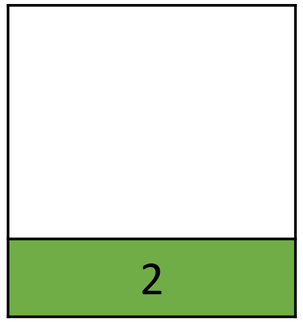
Data structures: Stack Example



push 2

push 7

Data structures: Stack Example

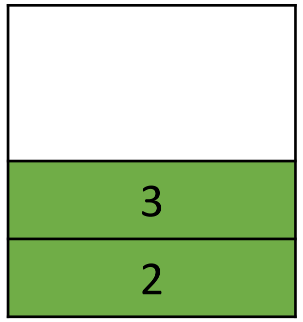


push 2

push 7

pop

Data structures: Stack Example



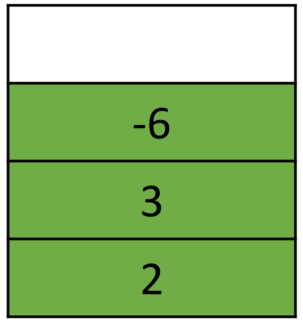
push 2

push 7

pop

push 3

Data structures: Stack Example



push 2

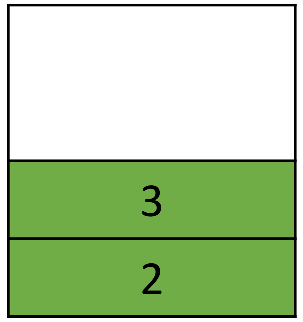
push 7

pop

push 3

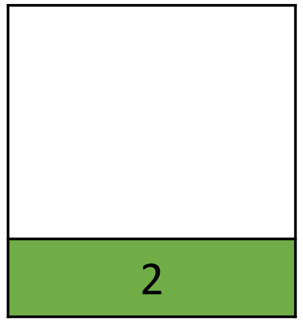
push -6

Data structures: Stack Example



push 2
push 7
pop
push 3
push -6
pop

Data structures: Stack Example



push 2

push 7

pop

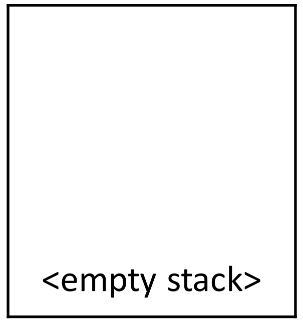
push 3

push -6

pop

pop

Data structures: Stack Example



push 2

push 7

pop

push 3

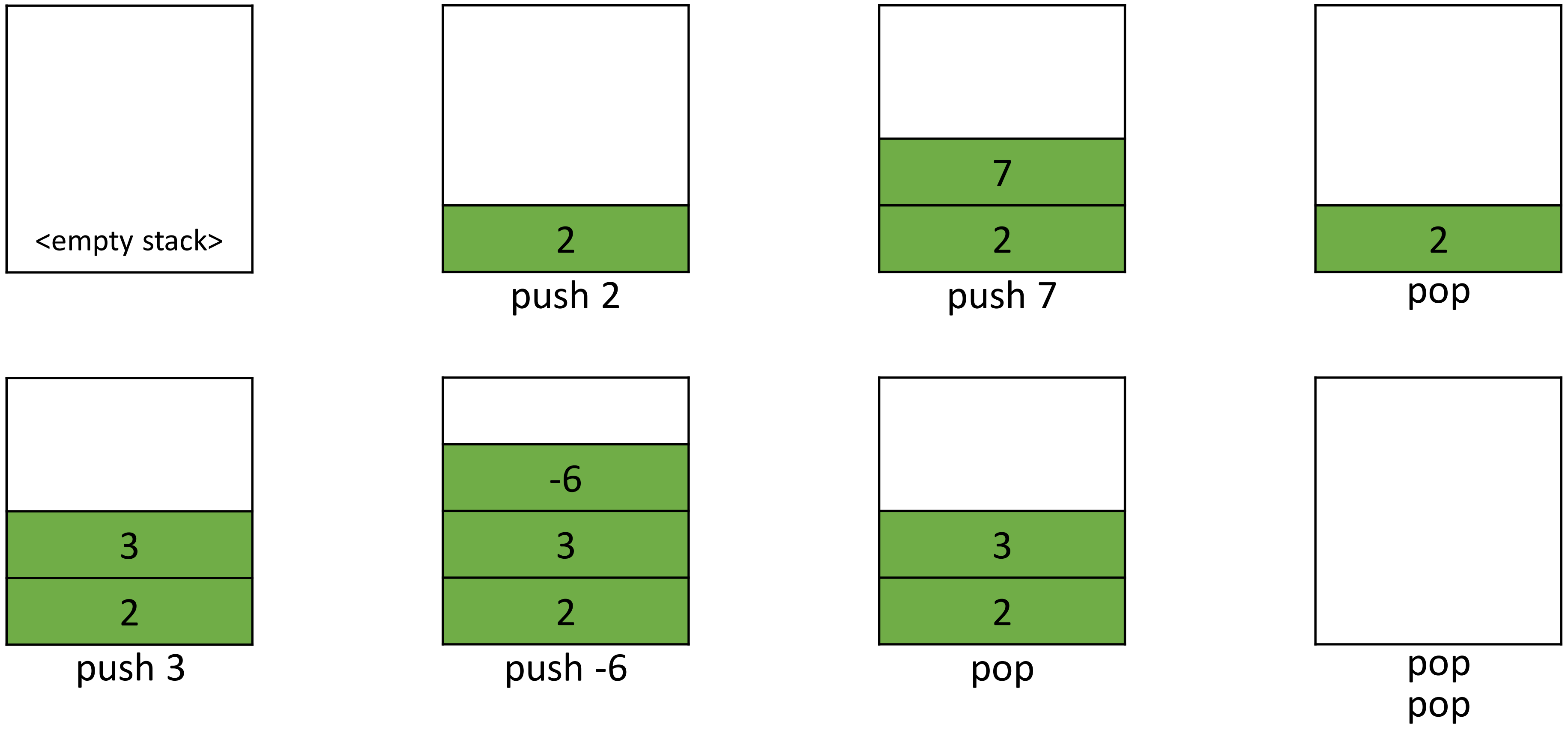
push -6

pop

pop

pop

Data structures: Stack Example



Data structures: Stack
Problem: Balanced parentheses

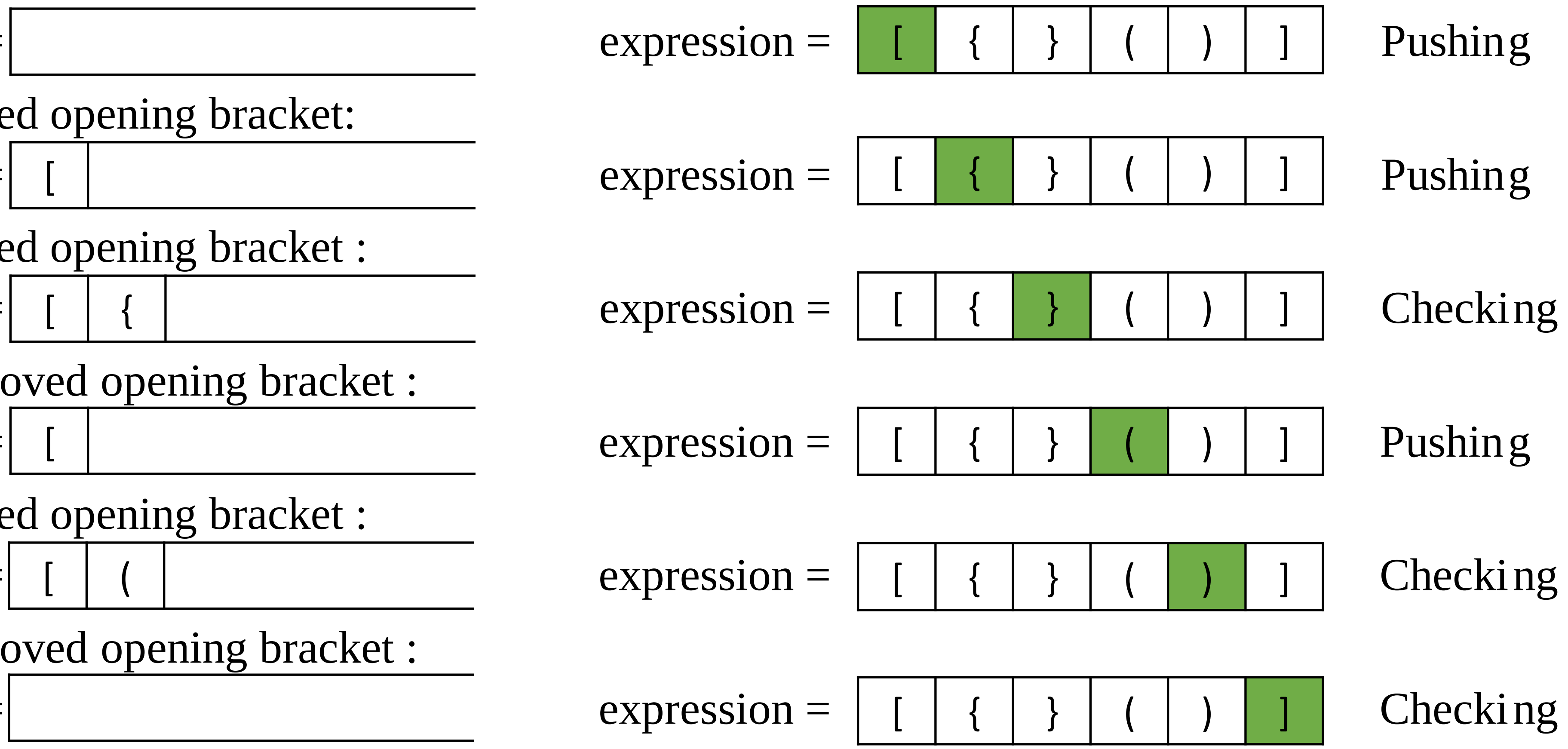
- Given an expression string ***exp***, develop an algorithm to examine whether the pairs and the orders of "{}()[]<>" are correct in ***exp***.
    - For example: "[()]<{<>[()()]()}>" is balanced.
    - For example: "[(])" is not balanced.

Data structures: Stack
Problem: Balanced parentheses

Algorithm:

- Declare a character stack S.
- Now traverse the expression string exp.
    - If the current character is a starting bracket (**‘(’ or ‘{’ or ‘[’ or ‘<’**) then push it to stack.
    - If the current character is a closing bracket (**‘)’ or ‘}’ or ‘]’ or ‘>’**) then pop from stack and if the popped character is the matching starting bracket then fine else parenthesis are not balanced.
- After complete traversal, if there is some starting bracket left in stack then “not balanced”.

Data structures: Stack
Problem: Balanced parentheses



- Initially:
  - stack =
- Step 1, added opening bracket:
  - stack = [
- Step 2, added opening bracket :
  - stack = [{
- Step 3, removed opening bracket :
  - stack = [
- Step 4, added opening bracket :
  - stack = [(
- Step 5, removed opening bracket :
  - stack =

expression = [{ } ( )] Pushing

expression = [{ } ( )] Pushing

expression = [{ } ( )] Checking

expression = [{ } ( )] Pushing

expression = [{ } ( )] Checking

expression = [{ } ( )] Checking