

Balanced Parentheses

```
PAIR = {'(': ')', '{': '}', '[': ']'}

# Complete the isBalanced function below.
def isBalanced(s):
    stack = []
    for char in s:
        if char in '({[':
            stack.append(char)
        else:
            if not stack or stack.pop() != PAIR[char]:
                return "NO"
    if stack:
        return "NO"
    return "YES"
```

We used two different *data structures* to implement this algorithm: a stack and a dictionary.

A stack is what we call a LIFO structure (Last In First Out). The common operations are push (which add a element to the top) and pop (which take the last element off the stack). [A stack is automatically implemented in python.](#)

To know exactly which parenthesis matches with which one, we need to remember the order, and therefore use a stack. Using another kind of structure would be ineficient if not impossible.

We also implement a dictionary, which we use to create pairs of parentheses. Using the dictionary, if we have one parenthesis, we know exactly which one we should have on the other side.