Quiz: Recursion (Practice Problems)

Note 1. You will have two quizzes on this material. The first quiz will be worth 2^2 points, and the second quiz worth 2^3 points. (Both will have 2^2 problems, the problems in the second quiz will be worth 2 points each.)

Note 2. All of these problems are designed to tests your ability to reason about recursion in an abstract setting. The problems in the `notes.py` file use recursion to actually implement the practical problem of binary search.

Note 3. Recall that recursion must have a base case. Without a base case, the recursion will cause a *stack overflow*, which in python is represented by throwing a `RuntimeError`. The code below catches this exception.

1 Basic Recursion

Problem 1. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ cat > foo.py <<EOF
def foo(xs):
    if len(xs) == 0:
        return 0
    return xs[0] + foo(xs[1:])
try:
    print('foo([1, 2, 3, 4, 5])=',foo([1, 2, 3, 4, 5]))
except RuntimeError:
    print('StackOverflow')
EOF
$ python3 foo.py
```

Problem 2. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ cat > foo.py <<EOF
def foo(xs):
    ret = foo(xs[1:])
    if len(xs) == 0:
        return 0
    return xs[0] + ret
try:
    print('foo([1, 2, 3, 4, 5])=',foo([1, 2, 3, 4, 5]))
except RuntimeError:
    print('StackOverflow')
EOF
$ python3 foo.py
```

Problem 3. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ cat > foo.py <<EOF
def foo(xs):
    return foo(xs[1:]) + xs[0]
try:
    print('foo([1, 2, 3, 4, 5])=',foo([1, 2, 3, 4, 5]))
except RuntimeError:
    print('StackOverflow')
EOF
$ python3 foo.py
```

Problem 4. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ cat > foo.py <<EOF
def foo(xs):
    if len(xs) == 0:
        return 0
    return - foo(xs[1:]) + xs[0]
try:
    print('foo([1, 2, 3, 4, 5])=',foo([1, 2, 3, 4, 5]))
except RuntimeError:
    print('StackOverflow')
EOF
$ python3 foo.py
```

Problem 5. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ cat > foo.py <<EOF
def foo(xs):
    if len(xs) == 0:
        return 0
    return foo(xs[1:-1]) * xs[-1]
try:
    print('foo([1, 2, 3, 4, 5])=',foo([1, 2, 3, 4, 5]))
except RuntimeError:
    print('StackOverflow')
EOF
$ python3 foo.py
```

Problem 6. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ cat > foo.py <<EOF
def foo(xs):
    if len(xs) == 2:
        return 0
    return foo(xs[1:-1]) + xs[-1]
try:
    print('foo([1, 2, 3, 4, 5])=',foo([1, 2, 3, 4, 5]))
except RuntimeError:
    print('StackOverflow')
EOF
$ python3 foo.py
```

Problem 7. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ cat > foo.py <<EOF
def foo(xs):
    ret = foo(xs[1:-1]) + xs[-1]
    if len(xs) == 2:
        return 0
    return ret
try:
    print('foo([1, 2, 3, 4, 5])=',foo([1, 2, 3, 4, 5]))
except RuntimeError:
    print('StackOverflow')
EOF
$ python3 foo.py
```

2 Using a Helper Function

Problem 8. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ cat > foo.py <<EOF
def foo(xs):
    def go(i):
        if i == len(xs):
            return 0
        return xs[i] + go(i+1)
    return go(0)
try:
    print('foo([1, 2, 3, 4, 5])=',foo([1, 2, 3, 4, 5]))
except RuntimeError:
    print('StackOverflow')
EOF
$ python3 foo.py
```

Problem 9. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ cat > foo.py <<EOF
def foo(xs):
    def go(i):
        if i == -1:
            return 0
        return xs[i] + go(i-1)
    return go(len(xs)-1)
try:
    print('foo([1, 2, 3, 4, 5])=',foo([1, 2, 3, 4, 5]))
except RuntimeError:
    print('StackOverflow')
EOF
$ python3 foo.py
```

Problem 10. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ cat > foo.py <<EOF
def foo(xs):
    def go(i):
        if i == 0:
            return xs[0]
        return xs[i] + go(i // 2)
    return go(len(xs)-1)
try:
    print('foo([1, 2, 3, 4, 5])=',foo([1, 2, 3, 4, 5]))
except RuntimeError:
    print('StackOverflow')
EOF
$ python3 foo.py
```

Problem 11. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ cat > foo.py <<EOF
def foo(xs):
    if len(xs) == 0:
        return 0
    def go(i):
        return xs[i] + go(i+1)
    return go(0)
try:
    print('foo([1, 2, 3, 4, 5])=',foo([1, 2, 3, 4, 5]))
except RuntimeError:
    print('StackOverflow')
EOF
$ python3 foo.py
```

3 With an accumulator

Problem 12. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ cat > foo.py <<EOF
def foo(xs):
    def go(i, acc):
        if len(xs) == i:
            return acc
        return go(i+1, acc + xs[i])
    return go(0, 0)
try:
    print('foo([1, 2, 3, 4, 5])=',foo([1, 2, 3, 4, 5]))
except RuntimeError:
    print('StackOverflow')
EOF
$ python3 foo.py
```

Problem 13. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ cat > foo.py <<EOF
def foo(xs):
    def go(i, acc):
        if len(xs) == i:
            return acc
        return go(i+1, acc + xs[i])
    return go(1, 2)
try:
    print('foo([1, 2, 3, 4, 5])=',foo([1, 2, 3, 4, 5]))
except RuntimeError:
    print('StackOverflow')
EOF
$ python3 foo.py
```

Problem 14. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ cat > foo.py <<EOF
def foo(xs):
    def go(i, acc):
        if len(xs) == i:
            return acc
        return go(i+1, - acc + xs[i])
    return go(0, 0)
try:
    print('foo([1, 2, 3, 4, 5])=',foo([1, 2, 3, 4, 5]))
except RuntimeError:
    print('StackOverflow')
EOF
$ python3 foo.py
```

4 Multiple Recursion

Problem 15. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ cat > foo.py <<EOF
def foo(xs):
    if len(xs) == 0:
        return 0
    return xs[0] + foo(xs[1:]) + foo(xs[2:])
try:
    print('foo([1, 2, 3])=',foo([1, 2, 3]))
except RuntimeError:
    print('StackOverflow')
EOF
$ python3 foo.py
```

Problem 16. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ cat > foo.py <<EOF
def foo(xs):
    if len(xs) == 0:
        return 0
    ret = xs[0]
    ret += foo(xs[1:])
    ret += foo(xs[:-1])
    return ret
try:
    print('foo([1, 2, 3])=',foo([1, 2, 3]))
except RuntimeError:
    print('StackOverflow')
EOF
$ python3 foo.py
```

Problem 17. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ cat > foo.py <<EOF
def foo(xs):
    def go(i, acc):
        if len(xs) <= i:
            return acc
        ret = 0
        ret += go(i+1, acc + xs[i])
        ret += go(i+2, acc + xs[i])
        ret += go(i+3, acc + xs[i])
        return ret
    return go(0, 0)
try:
    print('foo([1, 2, 3])=',foo([1, 2, 3]))
except RuntimeError:
    print('StackOverflow')
EOF
$ python3 foo.py
```

Problem 18. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ cat > foo.py <<EOF
def foo(xs):
    def go(i, acc):
        if len(xs) <= i:
            return acc
        ret = 0
        ret += go(i+1, acc + xs[i])
        ret += go(i+1, acc + xs[i])
        return ret
    return go(0, 0)
try:
    print('foo([1, 2, 3])=',foo([1, 2, 3]))
except RuntimeError:
    print('StackOverflow')
EOF
$ python3 foo.py
```

5 Building Lists with Recursion

Problem 19. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ cat > foo.py <<EOF
def foo(xs):
    if len(xs) == 10:
        return 1
    xs.append(1)
    return 2 * foo(xs)
try:
    print('foo([])=',foo([]))
except RuntimeError:
    print('StackOverflow')
EOF
$ python3 foo.py
```

Problem 20. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ cat > foo.py <<EOF
def foo(xs):
    if len(xs) == 10:
        return 1
    xs.append(1)
    return 2 * foo(xs)
try:
    print('foo([1, 2, 3])=',foo([1, 2, 3]))
except RuntimeError:
    print('StackOverflow')
EOF
$ python3 foo.py
```

Problem 21. Write the output of the final command in the following terminal session. If the command has no output, then leave the problem blank.

```
$ cd; rm -rf quiz; mkdir quiz; cd quiz
$ cat > foo.py <<EOF
def foo(xs):
    xs.append(1)
    return 2 * foo(xs)
    if len(xs) == 10:
        return 1
try:
    print('foo([1, 2, 3])=',foo([1, 2, 3]))
except RuntimeError:
    print('StackOverflow')
EOF
$ python3 foo.py
```