

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.

```
1 $ cd; rm -rf quiz; mkdir quiz; cd quiz
2 $ cat > foo.py <<EOF
3 def foo(xs):
4     if len(xs) == 0:
5         return 0
6     return xs[0] + foo(xs[1:])
7 try:
8     print('foo([1, 2, 3, 4, 5])=',foo([1, 2, 3, 4, 5]))
9 except RuntimeError:
10     print('StackOverflow')
11 EOF
12 $ 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.

```
1 $ cd; rm -rf quiz; mkdir quiz; cd quiz
2 $ cat > foo.py <<EOF
3 def foo(xs):
4     ret = foo(xs[1:])
5     if len(xs) == 0:
6         return 0
7     return xs[0] + ret
8 try:
9     print('foo([1, 2, 3, 4, 5])=',foo([1, 2, 3, 4, 5]))
10 except RuntimeError:
11     print('StackOverflow')
12 EOF
13 $ 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.

```
1 $ cd; rm -rf quiz; mkdir quiz; cd quiz
2 $ cat > foo.py <<EOF
3 def foo(xs):
4     return foo(xs[1:]) + xs[0]
5 try:
6     print('foo([1, 2, 3, 4, 5])=',foo([1, 2, 3, 4, 5]))
7 except RuntimeError:
8     print('StackOverflow')
9 EOF
10 $ 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.

```
1 $ cd; rm -rf quiz; mkdir quiz; cd quiz
2 $ cat > foo.py <<EOF
3 def foo(xs):
4     if len(xs) == 0:
5         return 0
6     return - foo(xs[1:]) + xs[0]
7 try:
8     print('foo([1, 2, 3, 4, 5])=',foo([1, 2, 3, 4, 5]))
9 except RuntimeError:
10     print('StackOverflow')
11 EOF
12 $ 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.

```
1 $ cd; rm -rf quiz; mkdir quiz; cd quiz
2 $ cat > foo.py <<EOF
3 def foo(xs):
4     if len(xs) == 0:
5         return 0
6     return foo(xs[1:-1]) * xs[-1]
7 try:
8     print('foo([1, 2, 3, 4, 5])=',foo([1, 2, 3, 4, 5]))
9 except RuntimeError:
10     print('StackOverflow')
11 EOF
12 $ 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.

```
1 $ cd; rm -rf quiz; mkdir quiz; cd quiz
2 $ cat > foo.py <<EOF
3 def foo(xs):
4     if len(xs) == 2:
5         return 0
6     return foo(xs[1:-1]) + xs[-1]
7 try:
8     print('foo([1, 2, 3, 4, 5])=',foo([1, 2, 3, 4, 5]))
9 except RuntimeError:
10     print('StackOverflow')
11 EOF
12 $ 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.

```
1 $ cd; rm -rf quiz; mkdir quiz; cd quiz
2 $ cat > foo.py <<EOF
3 def foo(xs):
4     ret = foo(xs[1:-1]) + xs[-1]
5     if len(xs) == 2:
6         return 0
7     return ret
8 try:
9     print('foo([1, 2, 3, 4, 5])=',foo([1, 2, 3, 4, 5]))
10 except RuntimeError:
11     print('StackOverflow')
12 EOF
13 $ 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.

```
1 $ cd; rm -rf quiz; mkdir quiz; cd quiz
2 $ cat > foo.py <<EOF
3 def foo(xs):
4     def go(i):
5         if i == len(xs):
6             return 0
7         return xs[i] + go(i+1)
8     return go(0)
9 try:
10     print('foo([1, 2, 3, 4, 5])=',foo([1, 2, 3, 4, 5]))
11 except RuntimeError:
12     print('StackOverflow')
13 EOF
14 $ 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.

```
1 $ cd; rm -rf quiz; mkdir quiz; cd quiz
2 $ cat > foo.py <<EOF
3 def foo(xs):
4     def go(i):
5         if i == -1:
6             return 0
7         return xs[i] + go(i-1)
8     return go(len(xs)-1)
9 try:
10     print('foo([1, 2, 3, 4, 5])=',foo([1, 2, 3, 4, 5]))
11 except RuntimeError:
12     print('StackOverflow')
13 EOF
14 $ 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.

```
1 $ cd; rm -rf quiz; mkdir quiz; cd quiz
2 $ cat > foo.py <<EOF
3 def foo(xs):
4     def go(i):
5         if i == 0:
6             return xs[0]
7         return xs[i] + go(i // 2)
8     return go(len(xs)-1)
9 try:
10     print('foo([1, 2, 3, 4, 5])=',foo([1, 2, 3, 4, 5]))
11 except RuntimeError:
12     print('StackOverflow')
13 EOF
14 $ 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.

```
1 $ cd; rm -rf quiz; mkdir quiz; cd quiz
2 $ cat > foo.py <<EOF
3 def foo(xs):
4     if len(xs) == 0:
5         return 0
6     def go(i):
7         return xs[i] + go(i+1)
8     return go(0)
9 try:
10     print('foo([1, 2, 3, 4, 5])=',foo([1, 2, 3, 4, 5]))
11 except RuntimeError:
12     print('StackOverflow')
13 EOF
14 $ 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.

```
1 $ cd; rm -rf quiz; mkdir quiz; cd quiz
2 $ cat > foo.py <<EOF
3 def foo(xs):
4     def go(i, acc):
5         if len(xs) == i:
6             return acc
7         return go(i+1, acc + xs[i])
8     return go(0, 0)
9 try:
10     print('foo([1, 2, 3, 4, 5])=',foo([1, 2, 3, 4, 5]))
11 except RuntimeError:
12     print('StackOverflow')
13 EOF
14 $ 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.

```
1 $ cd; rm -rf quiz; mkdir quiz; cd quiz
2 $ cat > foo.py <<EOF
3 def foo(xs):
4     def go(i, acc):
5         if len(xs) == i:
6             return acc
7         return go(i+1, acc + xs[i])
8     return go(1, 2)
9 try:
10     print('foo([1, 2, 3, 4, 5])=',foo([1, 2, 3, 4, 5]))
11 except RuntimeError:
12     print('StackOverflow')
13 EOF
14 $ 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.

```
1 $ cd; rm -rf quiz; mkdir quiz; cd quiz
2 $ cat > foo.py <<EOF
3 def foo(xs):
4     def go(i, acc):
5         if len(xs) == i:
6             return acc
7         return go(i+1, - acc + xs[i])
8     return go(0, 0)
9 try:
10     print('foo([1, 2, 3, 4, 5])=',foo([1, 2, 3, 4, 5]))
11 except RuntimeError:
12     print('StackOverflow')
13 EOF
14 $ 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.

```
1 $ cd; rm -rf quiz; mkdir quiz; cd quiz
2 $ cat > foo.py <<EOF
3 def foo(xs):
4     if len(xs) == 0:
5         return 0
6     return xs[0] + foo(xs[1:]) + foo(xs[2:])
7 try:
8     print('foo([1, 2, 3])=',foo([1, 2, 3]))
9 except RuntimeError:
10     print('StackOverflow')
11 EOF
12 $ 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.

```
1 $ cd; rm -rf quiz; mkdir quiz; cd quiz
2 $ cat > foo.py <<EOF
3 def foo(xs):
4     if len(xs) == 0:
5         return 0
6     ret = xs[0]
7     ret += foo(xs[1:])
8     ret += foo(xs[:-1])
9     return ret
10 try:
11     print('foo([1, 2, 3])=',foo([1, 2, 3]))
12 except RuntimeError:
13     print('StackOverflow')
14 EOF
15 $ 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.

```
1 $ cd; rm -rf quiz; mkdir quiz; cd quiz
2 $ cat > foo.py <<EOF
3 def foo(xs):
4     def go(i, acc):
5         if len(xs) <= i:
6             return acc
7         ret = 0
8         ret += go(i+1, acc + xs[i])
9         ret += go(i+2, acc + xs[i])
10        ret += go(i+3, acc + xs[i])
11        return ret
12    return go(0, 0)
13 try:
14     print('foo([1, 2, 3])=',foo([1, 2, 3]))
15 except RuntimeError:
16     print('StackOverflow')
17 EOF
18 $ 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.

```
1 $ cd; rm -rf quiz; mkdir quiz; cd quiz
2 $ cat > foo.py <<EOF
3 def foo(xs):
4     def go(i, acc):
5         if len(xs) <= i:
6             return acc
7         ret = 0
8         ret += go(i+1, acc + xs[i])
9         ret += go(i+1, acc + xs[i])
10        return ret
11    return go(0, 0)
12 try:
13     print('foo([1, 2, 3])=',foo([1, 2, 3]))
14 except RuntimeError:
15     print('StackOverflow')
16 EOF
17 $ 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.

```
1 $ cd; rm -rf quiz; mkdir quiz; cd quiz
2 $ cat > foo.py <<EOF
3 def foo(xs):
4     if len(xs) == 10:
5         return 1
6     xs.append(1)
7     return 2 * foo(xs)
8 try:
9     print('foo([])=', foo([]))
10 except RuntimeError:
11     print('StackOverflow')
12 EOF
13 $ 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.

```
1 $ cd; rm -rf quiz; mkdir quiz; cd quiz
2 $ cat > foo.py <<EOF
3 def foo(xs):
4     if len(xs) == 10:
5         return 1
6     xs.append(1)
7     return 2 * foo(xs)
8 try:
9     print('foo([1, 2, 3])=', foo([1, 2, 3]))
10 except RuntimeError:
11     print('StackOverflow')
12 EOF
13 $ 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.

```
1 $ cd; rm -rf quiz; mkdir quiz; cd quiz
2 $ cat > foo.py <<EOF
3 def foo(xs):
4     xs.append(1)
5     return 2 * foo(xs)
6     if len(xs) == 10:
7         return 1
8 try:
9     print('foo([1, 2, 3])=', foo([1, 2, 3]))
10 except RuntimeError:
11     print('StackOverflow')
12 EOF
13 $ python3 foo.py
```