# Exercise 7

Bookmark this page

## Exercise 7

4 points possible (graded)

**ESTIMATED TIME TO COMPLETE: 10 minutes**

Consider the following Python procedures. For each one, specify its order of growth.

1.
```
def lenRecur(s):
    if s == '':
        return 0
    else:
        return 1 + lenRecur(s[1:])
```

Select an option

2.

```
def isIn(a, s):
    '''
    a is a character, or, singleton string.
    s is a string, sorted in alphabetical order.
    '''
    if len(s) == 0:
        return False
    elif len(s) == 1:
        return a == s
    else:
        test = s[len(s)//2]
        if test == a:
            return True
        elif a < test:
            return isIn(a, s[:len(s)//2])
        else:
            return isIn(a, s[len(s)//2+1:])
```

Select an option

3.

```
def union(L1, L2):
    '''
    L1 & L2 are lists of the same length, n
    '''
    temp = L1[:]
    for e2 in L2:
        flag = False
        for check in temp:
            if e2 == check:
                flag = True
                break
        if not flag:
            temp.append(e2)
    return temp
```

For this problem, assume $n$ = len(L1) = len(L2)

Select an option

4.

```
def unionNew(L1, L2):
    '''
    L1 & L2 are lists of the same length, n
    '''
    temp = []
    for e1 in L1:
        flag = False
        for e2 in L2:
            if e1 == e2:
                flag = True
                break
        if not flag:
            temp.append(e1)
    return temp + L2
```

For this problem, assume $n$ = len(L1) = len(L2)

Select an option

Submit

## Exercise 7

**Topic:** Lecture 11 / Exercise 7

Show Discussion