6.00 Problem Set 7

Out: Friday, October 17
Due: Tuesday, October 21

This problem set is designed to help you solidify your understanding of some material that we have covered in lecture, but not emphasized on the programming problems. You should do it, but do NOT hand it in.

1) What is the computational complexity of fact0? Explain your answer.

```python
def fact0(i):
    assert type(i) == int and i >= 0
    if i == 0 or i == 1:
        return 1
    return i * fact0(i-1)
```

2) What is the computational complexity of fact1? Explain your answer.

```python
def fact1(i):
    assert type(i) == int and i >= 0
    res = 1
    while i > 1:
        res = res * i
        i -= 1
    return res
```

3) What is the computational complexity of makeSet? Explain your answer.

```python
def makeSet(s):
    assert type(s) == str
    res = ''
    for c in s:
        if not c in res:
            res = res + c
    return res
```

4) What is the computational complexity of intersect? Explain your answer.

```python
def intersect(s1, s2):
    assert type(s1) == str and type(s2) == str
    s1 = makeSet(s1)
    s2 = makeSet(s2)
    res = ''
    for e in s1:
        if e in s2:
            res = res + e
    return res
```
5) Present a hand simulation of the code below. Describe the value to which each identifier is bound after each step of the computation. Note that “s1” and “s2” exist in more than one scope.

```python
def swap0(s1, s2):
    assert type(s1) == list and type(s2) == list
    tmp = s1[:]
    s1 = s2[:]
    s2 = tmp
    return

s1 = [1]
s2 = [2]
swap0(s1, s2)
print s1, s2
```

6) Present a hand simulation of the following code:

```python
def swap1(s1, s2):
    assert type(s1) == list and type(s2) == list
    return s2, s1

s1 = [1]
s2 = [2]
s1, s2 = swap1(s1, s2)
print s1, s2
```

7) Present a hand simulation of the following code:

```python
def rev(s):
    assert type(s) == list
    for i in range(len(s)/2):
        tmp = s[i]
        s[i] = s[-(i+1)]
        s[-(i+1)] = tmp

s = [1,2,3]
rev(s)
print s
```