Big-O Analysis Practice:

```python
def bigOh1(n):
    x = y = 0
    while (x < n):
        while (y < n):
            print("y")
            y += 3
        print("x")
        x += 4
```

bigOh1 efficiency:

```python
def bigOh2(N):
    L = range(10**4)
    for x in range(0, 2**N, 2**N/N**2):
        if x in set(L):
            print ("beep boop")
        print("my heart is in the work")
```

bigOh2 efficiency:

Python

```python
def bigOh3(L):
    # assume L is an NxN (square) 2d list
    N = len(L)
    s = 0
    for row in L:
        a = sorted(row)
        s += 1
    return s
```

bigOh3 efficiency:

Python

```python
import string
def bigOh4(s):
    #s is a string of length N
    result = []
    for char in s:
        if char in string.ascii_lowercase:
            if char.upper() in s:
                result.append(s.count(char.upper()))
    return result
```

bigOh4 efficiency: