

CSC148 - Worst-Case Runtime Analysis

Give the worst-case big-oh time efficiency bound for each of the following functions.

```
1. def bigoh1(n: int) -> None:
    total = 0
    for i in range(100, n):
        total = total + 1

    print(total)
```

```
2. def bigoh2(n: int) -> None:
    total = 0
    for i in range(1, n // 2):
        total = total + 1
    for j in range(1, n * n):
        total = total + 1

    print(total)
```

```
3. def bigoh3(n: int) -> None:
    total = 0
    if n % 2 == 0:
        for j in range(1, n * n):
            total = total + 1
    else:
        for k in range(5, n + 1):
            total = total + k

    print(total)
```

```
4. def bigoh4(m: int, n: int) -> None:
    total = 0
    for i in range(1, n + 1):
        for j in range(1, m + 1):
            total = total + 1

    print(total)
```

5. Super Challenge

```
def bigoh5(lst: list) -> None:
    '''Pre: lst of length n contains integers 0 to n - 1
    in some order.'''
    for i in range(len(lst)):
        while lst[i] != i:
            lst[lst[i]], lst[i] = lst[i], lst[lst[i]]
```