COMP152 — SP25 — Lab 5
Loop and Big-O Worksheet

February 18, 2025

Loops: While and For

For each loop shown below:

- Determine *exactly* (either a literal value or a function of n) how many times the ... line will be executed.
- Determine the Big-0 characterization of the loop.
- Translate to the equivalent while/for loop. This means a loop with a counter (i,j,k, etc.) that takes on the exact same sequence of values, in the same order, repeats the same number of times, and has the same Big-O characterization.

1.
```
j = 2
while j < n:
    ...
    j += 3
```

2.
```
for i in range(0,n-1,n//3):
    ...
```

3.
```
k = n+5
while k >= 4:
    ...
    k -= 2
```

4.
```
for i in range(n):
    for j in range(n-1,i-1,-1):
        ...
```

5.
```
i = 0
while i<=(n-1):
   for j in range(1,n,2):
      ...
   i += 2
```

Big-O Analysis

Determine the Big-0 characterization of the worst case time for each of the following. Clearly identify/name the n for each.

1.
```
def foo(l : List[int]):
    s = 0
    for i in range(2**10):
        k = 1
        while k < len(l):
            s += l[k]
            k *= 2
    return s
```

2.
```
def foo(l : List[int], a : int, b : int):
    k = []
    for i in range(len(l)):
        if a < l[i] < b:
            k.append(l[i])
    return k
```

3.
```
def foo(l : List[int],a : int, b : int):
    i = 0
    while i < len(l):
       if a < l[i] < b:
          del l[i]
        else:
          i += 1
    return
```

4.
```
def foo(l : List[int], k : int):
    l.sort()
    return l[len(s)-k:]
```

5.
```
def foo(l : List[int]):
    for i in range(1,n):
        j = i
        while j>0 and l[j] < l[j-1]:
            l[j],l[j-1] = l[j-1],l[j]
            j -= 1
    return
```

6.
```
def foo(l : List[int]):
    i = 1
    while i < len(l):
        for j in range(0,len(l),i):
            l[j] = l[j]+l[j+2*i-1]
        i *= 2
    return
```