

Dounglan Cheung

3.4.2020

CIS351 HW6 Algorithm Analysis

Listing 1 Count all arithmetic operations.

```
public static int someFunc1(int[] numbers) {  
    int sum = 0;  
    for (int num : numbers) {  
        sum += num;  
        for (int i = 0; i < 20; i++) {  
            sum += i;  
        }  
    }  
    return sum;  
}
```

Answer:

- Input size: $N + 20N$
- BigO complexity: $O(N)$

Listing 2 Count all arithmetic operations.

```
public static int fun(int[] numbers) {  
    int sum = 0;  
    for (int i = 0; i < numbers.length; i++) {  
        for (int j = i; j < numbers.length; j++) {  
            sum += numbers[i] * numbers[j];  
        }  
    }  
    return sum;  
}
```

Answer:

- Growth function: quadratic, N^2
- BigO complexity: $O(N^2)$

Listing 3 Count all arithmetic operations.(pseudo code)

PROCEDURE DoStuff(numbers1, numbers2)

```
s <- 0  
FOR x IN numbers1 DO  
    FOR y IN numbers2 DO  
        IF x < y DO  
            RETURN 0  
        ELSE  
            s <- s + x  
        ENDIF
```

```

        ENDFOR
    ENDFOR
    FOR x IN numbers2 DO
        s <- s + x
    ENDFOR
    RETURN s
return sum;
}

```

Answer:

- Input size: $N^2 + N$
- BigO complexity: $O(N^2)$

Listing 4 Count all arithmetic operations.

```

public static int fun2(String sentence) {
    int[] counts = new int[sentence.length()];
    for (int i = 0; i < sentence.length(); i++) {
        for (int j = i; j < sentence.length(); j++) {
            if (sentence.charAt(i) == sentence.charAt(j)) {
                counts[i] += 1;
            }
        }
    }
    int howMany = 0;
    for (int count : counts) {
        if (count > 1) {
            howMany++;
        }
    }
    return howMany;
}

```

Answer:

- Input size: $N^2 + N$
- BigO complexity: $O(N^2)$
- Question: Have we chosen the best basic operation here? Can you think of a different choice that would simplify the analysis, but lead to the same BigO complexity class? Yes. No I cannot think of any.

Listing 5 Count all arithmetic operations.

```

public static int fun4(int[] numbers) {
    int sum = 0;
    for (int i = numbers.length - 1; i >= 1; i /= 2) {
        for (int j = 0; j < numbers.length / 2; j++) {

```

```
        sum++;  
    }  
}  
return sum;  
}
```

Answer:

- BigO complexity: $O(\log n)$