School of
Information Systems

SMU
SINGAPORE MANAGEMENT UNIVERSITY

COR-IS1702: COMPUTATIONAL THINKING
WEEK 3: COMPLEXITY

Solutions to In-Class Ex

2020/21 Term 1

In-Class Exercises: What is the Big O?

f(n)	Big O
$3n^3 - 27n^2 + 9n + 10$	$O(n^3)$
$n^2 - \log n + 9n$	$O(n^2)$
$n \log n + 9n$	$O(n \log n)$
$2^n + n^2$	$O(2^n)$

In-Class Exercises

What is the Big O complexity?

Given an array `a` of `n` numbers, where `n > 10`, find out which of the first `10` numbers is the largest.

```
def findMaxTen(a)
    max = a[0]:
    for i in range(1,10):
        if max < a[i]:
            max = a[i]
    return max
end
```

- Number of 'comparison' operations is 9
  Complexity is O(1)

In-Class Exercises

What is the Big O complexity?

Given an array `a` of `n` numbers, find the smallest difference between any two numbers in the array `a`.

```
def findMinDiff(a)
    mindiff = (a[0]-a[1]).abs
    for i in 0 .. (a.length-2)
        for j in (i+1) .. (a.length-1)
            diff = (a[i]-a[j]).abs
            if mindiff > diff
                mindiff = diff
            end
        end
    end
    return mindiff
end
```

- Number of 'comparison' operations is n(n-1)/2
- Number of 'assignment' operations is at most n(n-1) + 1
- Number of 'subtraction' operations is n(n-1)/2 + 1
- Total number of operations is at most 2n(n-1) + 2
- Complexity is $O(n^2)$

In-Class Exercises

What is the Big O complexity?

There are `n` students in the class. Find 3 students with different last names.

- For the brute force approach, in the worst case, we have to inspect all possible groups of 3 students.
- There are ${}^{n}C_{3}$ possible groups of 3 students.

$$ {}^{n}C_{3} = \frac{n \times (n-1) \times (n-2)}{1 \times 2 \times 3} = \frac{n^3 - 3n^2 + 3n}{6} $$

- The order of complexity is $O(n^3)$.
  - The complexity is cubic with respect to `n`.