## Algorithms

# Multivariate Time Complexity Worksheet

Oftentimes the algorithms we use are for inputs of more than one dimension, multiple inputs, or both. Thus the time complexities we use to analyze the algorithm may be a multivariate expression.

1. Derive the time complexity for the following algorithm:

**Algorithm 1:** Algorithm to find the number of instances of "a" in a list of strings

**Data:** *strings* A list of $n$ lower-case strings that are all of length $k$
**Result:** *num_a* the number of a's in all the strings

```
1 num_a = 0;
2 for string ∈ strings do
3     for letter ∈ string do
4         if letter = "a" then
5             num_a+=1;
```

**SOLUTION:** $O(kn)$

2. Derive the time complexity for the following algorithm:

**Algorithm 2:** FOO$(m, n)$ counts the number of times it is called

**Data:** $m$ an integer, $n$ an integer
**Result:** an integer

```
1 FOO(m, n)
2 if n=0 then
3     return 1
4 sum = 0;
5 for 1≤ i ≤ m do
6     sum+=FOO(i, n − 1)
7 return sum
8
```

**SOLUTION:** I beleive the time complexity for this algorithm is $O(m^n)$ because there will be $n$ layers, and in each layer, the function is called about $m$ times,