

---

## DSC 40B - Discussion 02

---

### Problem 1. (Lab Problems)

- a) Let  $f(n) = (n + 1)(n - 1) \times 2^{\log_2(3^n)}$ . Which of the following asymptotic bounds on  $f$  is true?
- b) True or False. If  $f_1 = \Theta(g_1(n))$  and  $f_2 = \Omega(g_2(n))$  then  $\frac{f_1}{f_2} = \Theta(\frac{g_1}{g_2})$
- c) What is the worst case time complexity of the following function?

```
def foo(arr):  
    ''' arr is an array of size n'''  
    for x in arr:  
        for y in arr:  
            if (x+y) == 5:  
                return sum(arr)  
    return False
```

### Problem 2.

State the growth of the function below using  $\Theta$  notation, and prove your answer by finding constants which satisfy the definition of  $\Theta$  notation.

$$f(n) = \frac{n^3 - n^2 + n + 1000}{(n - 1)(n + 2)}$$

### Problem 3.

Consider the algorithm below.

```
def bogosearch(numbers, target):  
    """search by randomly guessing. `numbers` is an array of n numbers"""  
    n = len(numbers)  
  
    while True:  
        # randomly choose a number between 0 and n-1 in constant time  
        guess = np.random.randint(n)  
        if numbers[guess] == target:  
            return guess
```

We will set up the analysis of the expected time complexity of this algorithm.

- a) What are the cases? How many are there?
- b) What is the probability of case  $\alpha$ ?
- c) What is the running time in case  $\alpha$ ?

### Problem 4.

Provide a tight theoretical lower bound for the problems given below. Provide justification for your answer.

- a) Given an array of  $n$  numbers, find the sum of the numbers in the array.

**b)** Given a sorted array of  $n \geq 2$  numbers, find the second largest number in the array.