
DSC 40B - Discussion 01

Problem 1.

For each of the following pieces of code, state the time complexity using Θ notation.

```
a) def f_1(n):
    for i in range(1_000_000, n):
        for j in range(i):
            print(i, j)

b) def f_2(n):
    j = 0
    for i in range(n, n**5):
        while j < n:
            print(i, j)

c) def f_3(n):
    j = 1
    while j <= n:
        j = j * 2

d) def f_4(arr):
    """arr is an array of size n"""
    t = 0
    for i in range(n):
        t += sum(arr)

    for j in range(n**3):
        print(j//t)
```

Solution:

For explanations on each of these problems, check out the discussion recording [here](#). The timestamps for the question explanations are listed for each problem.

a) $f_1(n)$: $\Theta(n^2)$ \rightarrow 7 : 28

b) $f_2(n)$: $\Theta(\infty)$ \rightarrow 12 : 41

c) $f_3(n)$: $\Theta(\log_2 n)$ \rightarrow 16 : 29

d) $f_4(n)$: $\Theta(n^3)$ \rightarrow 19 : 50

```
def f_2_typo_fix(n):
    for i in range(n, n**5):
        j = 0
        while j < n:
            print(i, j)
            j += 1
```

b*) $f_2(n)$: $\Theta(n^6)$ \rightarrow 13 : 52

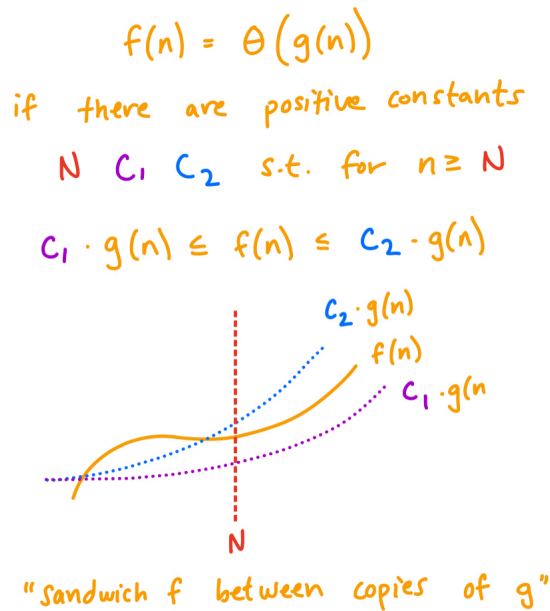
Problem 2.

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

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

Solution:

This question has infinitely many solutions. One such solution is shown below. Check out the [discussion](#) at 23 : 16 for an explanation.



Strategy : Guess and Check

Claim : $f(n) = \Theta(n)$

Prove : $c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)$

Upper Bound : guess $c_2 = 2 \rightarrow$ solve for N

$$f(n) \leq c_2 \cdot g(n)$$

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

$$n^3 - n^2 + n + 1000 \leq 2n^3 - 2n^2 - 4n$$

$$n^3 \geq n^2 + 5n + 1000$$

$$n \geq 11$$

Lower Bound : guess $c_1 = 0.5 \rightarrow$ solve for N

$$c_1 \cdot g(n) \leq f(n)$$

$$0.5n \leq \frac{n^3 - n^2 + n + 1000}{(n-1)(n+2)}$$

$$0.5n^3 - 0.5n^2 - n \leq n^3 - n^2 + n + 1000$$

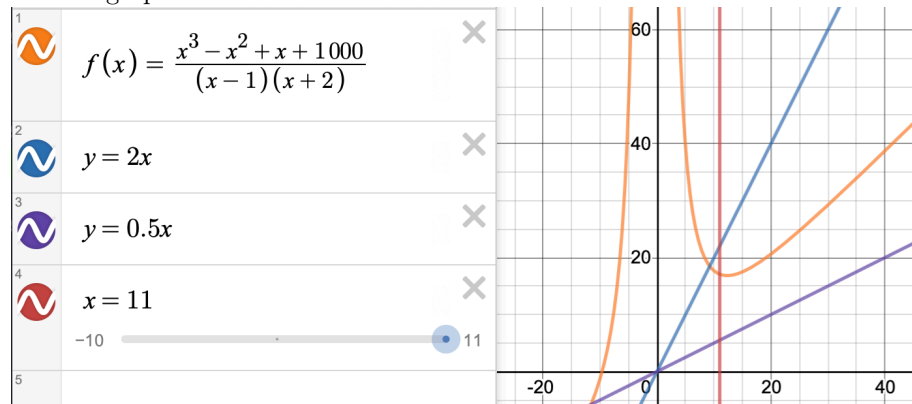
$$0.5n^3 \leq 0.5n^2 + 1000$$

$$n \geq 0$$

We have $n \geq 11$ and $n \geq 0$. Picking the MOST restrictive bound between these two options results in $N = 11$.

Therefore we have proved $f(n) = \Theta(n)$ with constants $N = 11, c_1 = 0.5, c_2 = 2$.

We can graph this to see that we are correct.



Problem 3.

Let $f(n) = \sum_{p=0}^n 3^p$. What is f in Θ notation?

Solution:

Check out the [discussion](#) at 41 : 30 for an explanation.

General form of a geometric sum $\sum_{p=0}^n x^p = \frac{1 - x^{n+1}}{1 - x}$.

Substituting our equation yields $\sum_{p=0}^n 3^p = \frac{1 - 3^{n+1}}{1 - 3}$.

Therefore, $f(n) = \Theta(3^n)$ after throwing out the constants.