1. Introduction

Unit 2, big theta notation. Concept to talk about the running time of algorithms.

1. Quiz: Divisible By Five

Tools for analyzing growth rates. Graph networks!

numbers = [361, 636, 277, 129, 434, 577, 796, 596, 727, 566, 156, 109, 714, 716, 546, 979, 366, 766, 137, 243, 331, 999,

922, 304, 657, 314, 634, 303, 677, 597, 363, 174, 431, 193, 361, 677, 403, 926, 279, 692, 749, 401, 346, 202,

763, 314, 333, 244, 796, 697, 674, 651, 517, 349, 337, 667, 617, 464, 379, 793, 542, 464, 962, 146, 946, 199,

302, 699, 606, 126, 519, 203, 137, 517, 146, 724, 696, 699, 747, 663, 126, 247, 469, 953, 396, 502, 562, 647,

364, 214, 346, 646, 331, 426, 763, 291, 557, 764, 939, 656, 753, 561, 797, 224, 537, 361, 263, 493, 196, 162,

362, 102, 629, 936, 663, 279, 966, 241, 907, 677, 945, 416, 122, 563, 667, 394, 654, 592, 977, 177, 666, 199,

463, 561, 954, 924, 991, 363, 754, 754, 199, 451, 796, 566, 629, 651, 517, 167, 704, 749, 622, 299, 466, 559,

973, 243, 639, 276, 603, 753]

reduced\_numbers = list()

for number in numbers:

reduced\_numbers.append(number % 10)

reduced\_product = reduced\_numbers[0]

for x in range(1, len(reduced\_numbers)):

product = reduced\_product \* reduced\_numbers[x]

reduced\_product = product % 10

if reduced\_product % 5 == 0:

print('YES, this product is divisible by 5.')

else:

print('NO, the product is not divisible by 5.')

# YES, this product is divisible by 5.

# trick = one value has to be divisible by 5

1. Quiz: Chain Network

m = n – 1 , m edges with n nodes if it is a chain network

1. Ring Network

‘complete the loop’ of a chain network

m=5

n=5

1. Quiz: Grid Network

Connectivity pattern of a grid/street-wise if it was a city.

n=20

m=15+16=31

If we have 256 nodes arranged as a square grid, how many edges do we have?

Square root of 256 = 16

16 \* 16 grid

n=256

m=15\*16\*2=480

1. Big-Theta

2n-2rad(n) is in Big Theta of n

Basically, after a threshold of n\_0, f(n) lies between c\_1 g(n) and c\_2 g(n). Think boundaries around the run time f(n).

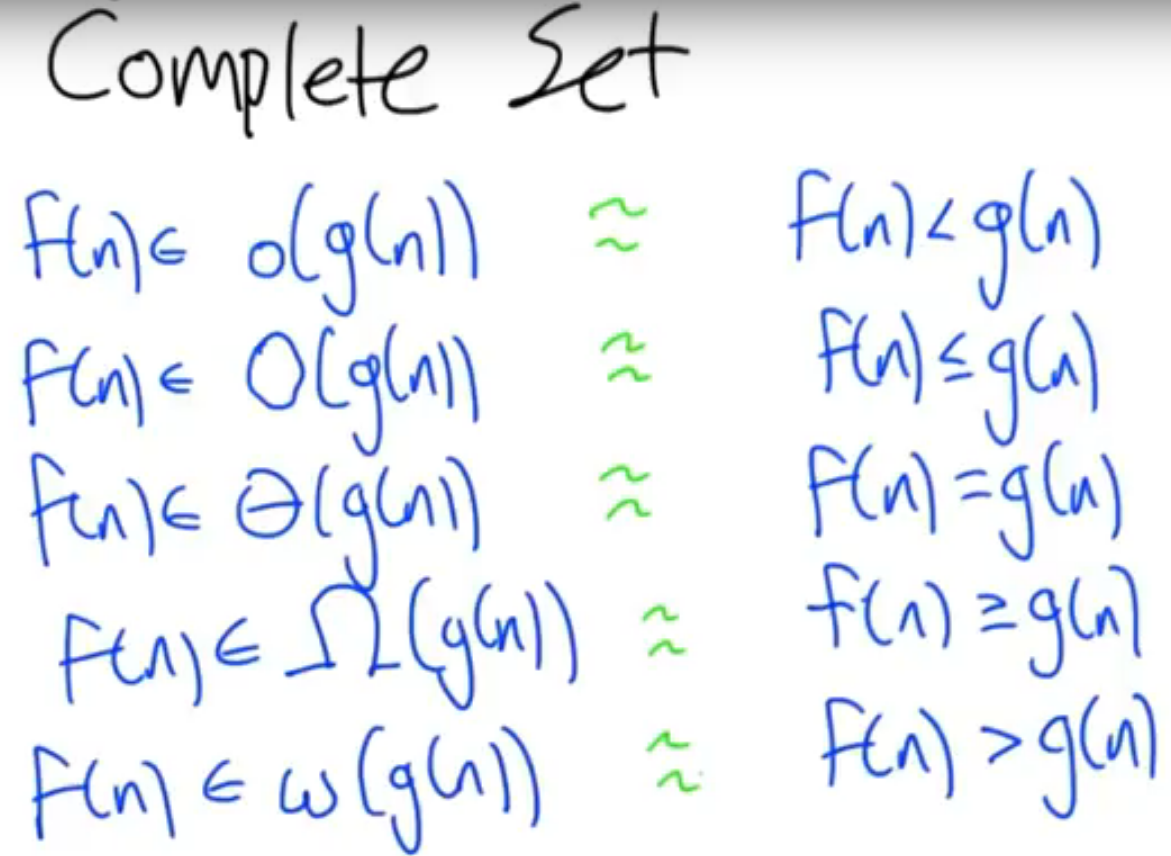
1. Quiz: Big-Theta Reflexive

Yes, use reciprocals to sandwich g by f.

1. Big-Theta Examples

Big Theta allows us to take complicated equations and represent them in a less complicated manner.

1. Other sets of Functions



1. Quiz: Big-Theta Practice

Itself, and the n^2 functions are approximately big theta.

1. Planar Graphs

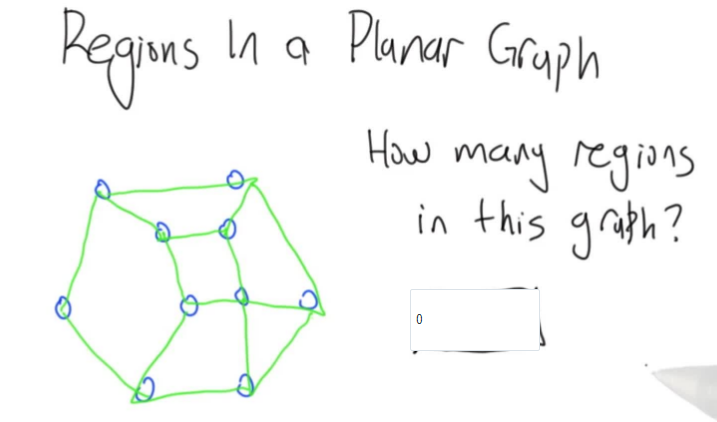
Edges do not cross.

1. Nodes, Edges, Regions

Euler returns, regions in a graph plus the region outside of the graph.

n-m+r=2

1. Quiz: Regions In A Planar Graph



7 regions

10-15+7=2

1. Eulers Formula

n-m+r=2

single node = 1-0+1=2 Base Case

add edge and node cancels for n-m+r=2

add edge alone = n-m+r=2

Inductions holds!

1. Growth Rates

m <= 3n – 6 in theta of n, therefore linear growth based off number of nodes.

1. Quiz: Complete Graph

m = Big-Theta of n

Not always the case, like in the complete graph.

def clique(n):

# Return the number of edges

# Try to use a mathematical formula...

edges = sum(range(0, n))

return edges

for x in range(6):

print(str(x) + '\t' + str(clique(x)))

n\*(n-1)/2 = quadratic

1. Hypercube

n: power of 2

n = 4 = 2^2, m = 4

Connect every node that differs by one from their binary bit pattern.

1. Quiz: Hypercube Edges

nlogn bits = 2 = log n

degree = logn from the edges

1. Quiz: Tree Graphs

Trees are connected graphs without cycles.

1. Randomly Generated Graphs

Erdos-Renyi Model

n nodes, p connectivity probability

generate n nodes, each node pair i,j has a randomly connected probability

1. Quiz: Recursive Graphs

It will generate a tree, noticeable at n=8 and greater.

1. Quiz: Recurrence Relation

Log n depth with n leaves.

1. Number Of Edges

2n = Big Theta of n

1. N Squared

T(n) = 2T(n/2) + Big Theta(n^2)

1. Quiz: Tangled Hypercube

None of these!

1. Recap