James Stallkamp

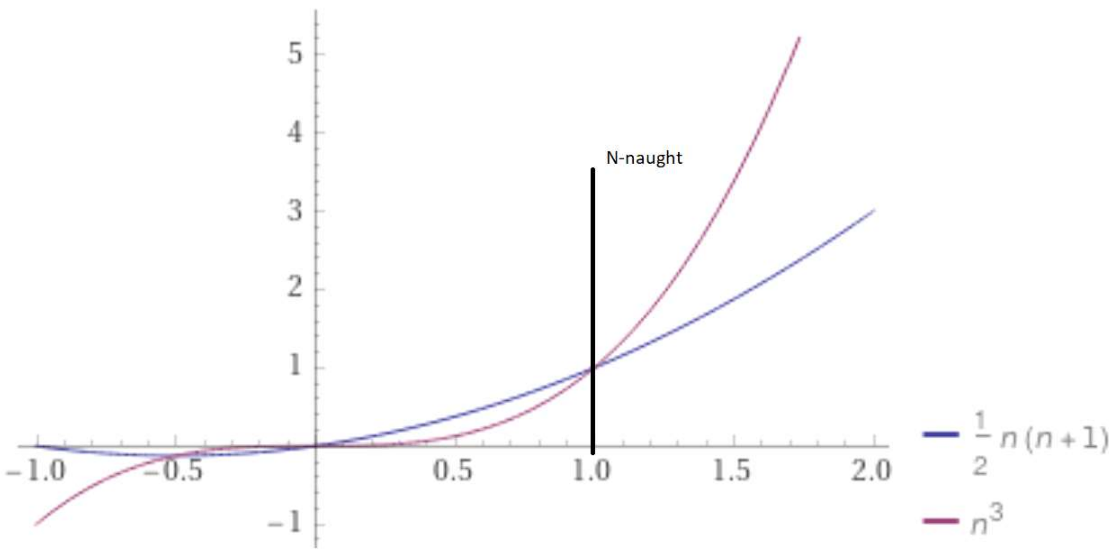
CS 325 Assignemt 1

1:A n(n+1)/2 ∈ O(n^3) Answer: True. n(n+1)/2 has a the same or lower order of growth aproximatly where n-naught is shown on the graph.

n(n+1)/2 = 1/2n^2 + 1/2n multiply "1/2n" term by n and combine = n^2

n^2 <= n^3

so n-naught = 1 and c = 1



1:B n(n+1)/2 ∈ Θ(n^2) Answer: True.

n(n+1)/2 = 1/2n^2 + 1/2n multiply "1/2n" term by n and combine = n^2

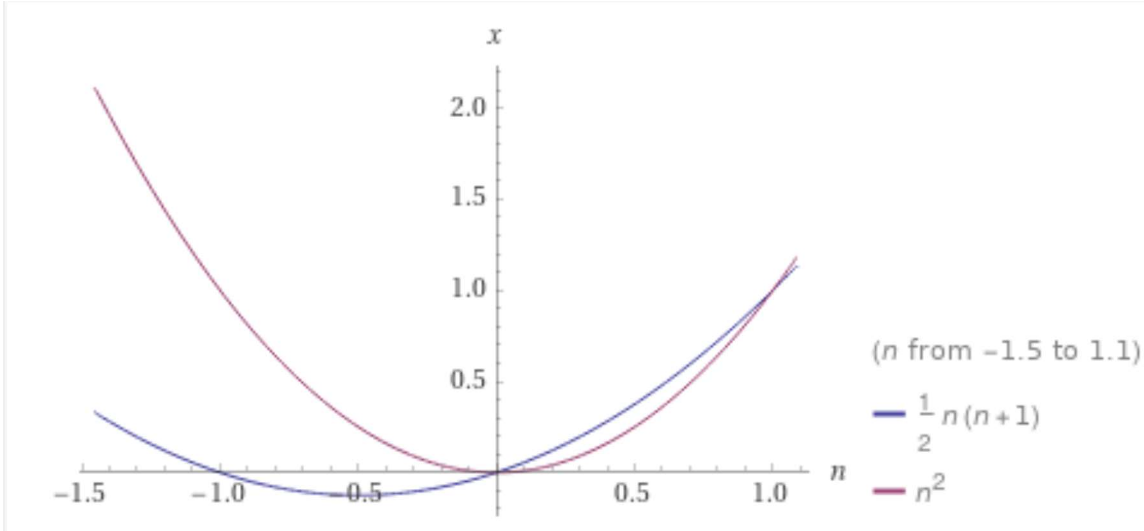
n^2 <= n^2 exactly

limit of (n(n+1)/2)/(n^2) , apply l hospitals

limit of (n+1/2)/n apply again

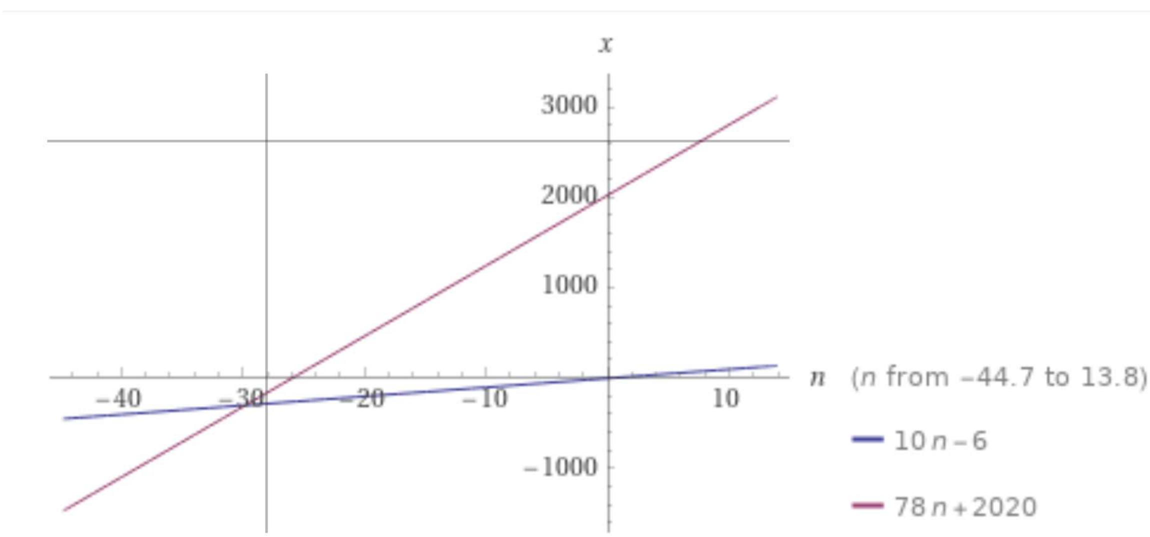
limit of (1/2) / 1 = 1/2 so a constant

implies that n(n+1)/2 ∈ Θ(n^2)



1:C 10n-6 ∈ Ω(78n + 2020) False.

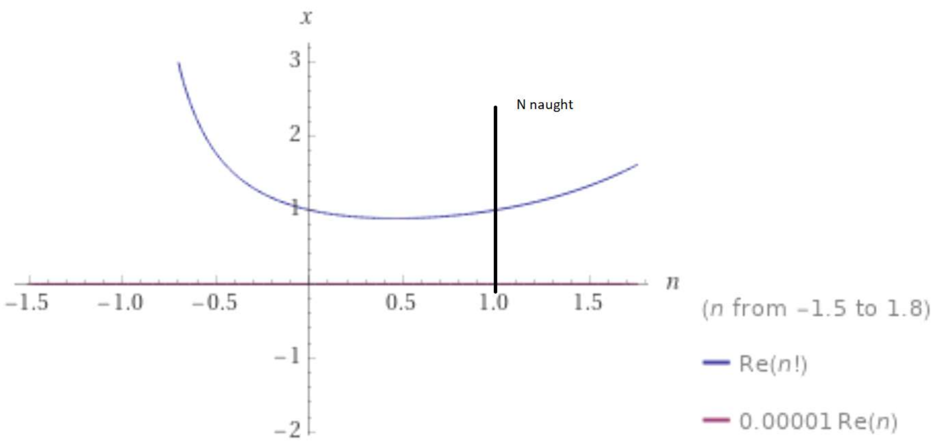
limit 10n-6/78n+2020 = 10/78 which is a constant so these functions have the same order of growth.



1:D n! ∈ Ω (0.00001n) True

lmt n->inf n! / .00001n = n*(n-1)*(n-2)..../.00001n = (n-1)*(n+2)..../.0002 applying limit is infinity

which implies that n! ∈ Ω (0.00001n)



2:A This function returns the difference between the max value and the min value in the array.

2:B The line "for i = 1 to n-1:"

2:C N-1 times. The loop goes from 1 to n-1 but has to cycle one more time for the value where it skips over the loop. So add 1 and get 1 to n or n-1

2:D O(n)

3:A The loop invariant is the sub array [i+1:j-1] has been reversed compared to the original array

3:B

Example n = 6

Base Case: begining of the loop. i = 2 and j = 3 i+1>j-1 so array would be reversed of the original

Inductive case: Assume loop invarint is true for ith iteration. Assuming at the ith iteration the current array from [Ai+1:Aj-1] is already reversed. The body of the loop will swap the current i and j elements outside of the currently reversed array making a now [Ai:Aj] reversed array. Then i and j are decremented and incremented respectfully which means at the end of each loop cycle the loop invariant holds true.

Termination: When the loop terminates i = -1 and j = n this combined with the loop invariant tells us that the entire array from A0:An has been fully reversed.