

from time import sleep blah = "" For this document, imagine that your computer moves at roughly the same speed as it is working, regardless what it is working on. We imported sleep so that each step in the process will take roughly 1/10 of a second, so you can see the process slowed down. (Long enough for you to get bored and sleepy when the process takes really long, this way you notice how longer scaling results in things that might not be desirable if speed is your top priority). Run the document (you can copy paste into your IDE if you want) and see how each function takes a different amount of time. "" def constantFunc(n): sleep(0.2) print(n) def linearFunc(n): for i in n: sleep(0.2) print(i) def exponentialFunc(n): for i in n: sleep(0.2) print(f'This is loop number {i}') for j in n: sleep(0.2) print(j) print('Constant function:') constantFunc([1,2,3]) constantFunc([1,2,3,4,5,6,7,8,9,10]) print('Linear function:') linearFunc([1,2,3]) linearFunc([1,2,3,4,5,6,7,8,9,10]) print('Exponential function:') exponentialFunc([1,2,3]) print('Longer list takes much longer!!! Imagine 100 items!') exponentialFunc([1,2,3,4,5,6,7,8,9,10]) # Logarithmic function is our phonebook example, aka a binary tree # We also could do factorial but that would take so long I'd nap and come back so we'll pass # An example of a factorial would be our recursive methods that we did a document or two ago