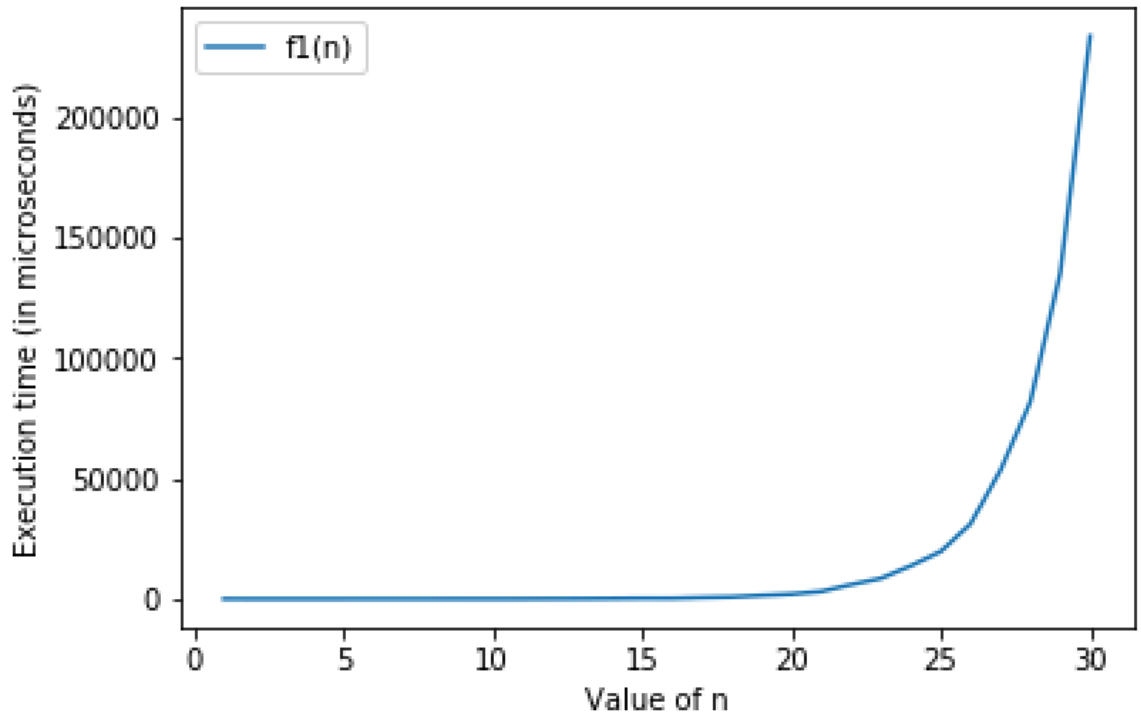
Name: Rishabh Jain
UID: 3035453608

Environment used:
Python 3.6.5 for MacOS (Version MacOS 10.13.6)
Programming done on Spyder (Python 3.6)

Q1: Plot a chart showing the actual running time of first version (exponential time version) on different values of *n*. Note that you cannot use too large values of *n*, otherwise, it will take a long long time for the program to run, you may not be able to get the result.

A1:
Graph has been plotted using matplotlib version 2.2.2
The 1st algorithm (f1(n)) has complexity $O(2^n)$
Thus the runtime increases with increase in n (with exponential relation)
Because of this exponential relation, a large value of n cannot be used as it be infeasible.



Values of n have been checked from 1 to 30 and plotted.
X axis shows values in microseconds (10^{-6} s)

Q2: Plot a chart showing the actual running times of the other two versions together in the same graph.

A2:
Graph has been plotted using matplotlib version 2.2.2
The 2nd algorithm (f2(n)) has complexity O(n). Thus, the execution time increases linearly with increase in n.
The 3nd algorithm (f3(n)) has complexity O(logn). Thus, the execution time has a logarithmic increase relation with increase in n.

In order to see the relative pattern of the two algorithms, the values of n used: (100000, 200000, 300000, … , 1000000)
X axis shows values in microseconds (10^{-6} s)