Question 1

from random import randint (1)

def shuffle(usuableList, number, newList): (n)

if len(usuableList) == 1: (n)

newList.append(usuableList[0]) (1)

return newList (1)

else: (n)

number = randint(0, len(usuableList)-1) (n)

newList.append(usuableList[number]) (n)

usuableList.remove(usuableList[number]) (n)

return shuffle(usuableList, number, newList) (n)

inputList = input(“Please enter a series of numbers”) (1)

usuableList = inputList.split(“,”) (1)

inputBool = False (1)

i = 0 (1)

while inputBool == False or i <= len(usuableList)-1: (n)

try: (n)

int(usuableList[i]) (n)

i = i + 1 (n)

inputBool = True (n)

except ValueError: (n)

print(“Please only input a series of integers”) (n)

inputList = input(“Please enter a series of numbers”) (n)

usuableList = inputList.split(“,”) (n)

i = 0 (n)

number = 0 (1)

newList = [] (1)

print(shuffle(usuableList, number, newList)) (1)

Question 1 runtime – 17n +10

Big O (n)

Question 2

def division(fiveCount, number): (log(n))

if number >= 0 and number < 1: (log(n))

return fiveCount (1)

else: (log(n))

hold = fiveCount + number/5 (log(n))

fiveCount = int(hold) (log(n))

return division(fiveCount, number/5) (log(n))

inputNumber = input("Please enter a number") (1)

floatHolder = float(inputNumber) (1)

count = 0 (1)

print(division(count, floatHolder)) (1)

Question 2 run time – 6Log(n)+5

Log5(n) as you divide n the number of times given to you by Log5(n)

Big O – O(log(n))

Question 3

numberOfEggs = [] (1)

numberOfAliens = [] (1)

numberOfAliens.append(int(input("Please enter amont of aliens that have landed"))) (1)

numberOfEggsLaid = int(input("Please enter the amount laid by an alien each day")) (1)

numberOfDaysPerHatch = int(input("Please enter amount of days for an egg to hatch")) (1)

numberOfEggs.append(numberOfEggsLaid) (1)

days = int(input("Number Of days the aliens are invading")) (1)

numberHatched = 0 (1)

for i in range(1,days): (n)

if i-numberOfDaysPerHatch >=0: (n)

numberHatched = numberOfEggs[i-numberOfDaysPerHatch] (n-m)

numberOfAliens.append(numberOfAliens[i-1] + numberHatched) (n-m)

else: (m)

numberOfAliens.append(numberOfAliens[i-1]) (m)

numberOfEggs.append(numberOfAliens[i] \* numberOfEggsLaid) (n)

print(numberOfAliens) (1)

print(numberOfAliens[len(numberOfAliens)-1], "Aliens") (1)

Question 3 runtime = 3n + 2m + (2n-2m) + 10

Big O – O(n)

Depends on hatching days too as some parts of code run for the length of m and the other part of the if statement runs for n-m. However as n is the most costly, for example as the number of days inputted is the larger number between n and m, as otherwise no eggs would hatch, it thus determines how many times the majority of the code runs. Therefore the Big O value is n.