Name: Steven Qian

Computing ID: uen9ke

Homework Name: HW5-BigOh

Resources Used: <https://www.cuemath.com/sum-of-squares-formula/> Used it for formula of sum of squares.

Answers

2.1

Although P1 sorts faster than P2 at a low number of inputs (20), P1 might not necessarily be faster than P2 at much higher order of magnitudes of inputs.

2.2

If the number of inputs is 10000 and P1 is still faster than P2, this also cannot prove that P1 will sort faster than P2 on the million names phonebook. However, if P2 now sorts faster than P1, we can conclude that P1 has a higher growth rate than P2 in the long run, and P2 sorts faster than P1 at a higher number of inputs. In this second case, P2 will sort faster than P1 on the million names phonebook.

2.3

Since Igor's code is clear and well-commented, we should read the source code first to determine which algorithm is faster at millions of inputs. After that we can also run tests at lower number of inputs to see if one algorithm beats another as inputs increase, but this might not necessarily work all the time at low number of inputs.

2.4

1) C

2) B

3) A

4) E

3.1

E

3.2

H

3.3

A

3.4

J

4.1

For n > 7, because two algorithm reach same time between n=7 and n=8

4.2

O(n lg n)

4.3

a) g(n) = O(f(n)) (f(n) grows faster than g(n))

b) f(n) = O(g(n)) (g(n) grows faster than f(n))

4.4

Lowest

5 lg(n + 100)^10

(ln n)^2

n^(1/3)

0.001n^4 + 3n^3 + 1

(n-2)!

2^(2n)

3^n

Highest

4.5

a)

It calculates the sum of the squares of integers from 0 to n.

b)

The basic operation is take the result of i times i and then add it to the variable s

c)

This program executes n+1 times.

d)

O(n)

e)

Using the fact that sum of squares of positive integers is (n \* (n + 1) \* (2n + 1)) / 6, we can plug n into this equation to get the result. This way the complexity of the algorithm will be reduced to O(1).