CSC 140 Advanced Algorithm Design and Analysis
CSC 206 Algorithms and Paradigms
Spring 2022
Assignment 1: Asymptotic Complexity
Due Monday February 14, 2022

Question 1 [30 Points]

For each pair of functions below, first determine which **function grows asymptotically faster**, and then express the relation between the two functions using all the asymptotic notations (**θ, O, Ω, o** and **ω)** that apply. Only give the relations in which **f(n)** appears on the left-hand side. Prove the relation between the **first two pairs** of functions using the limit definition.

(1) f(n) = n logn   g(n) = n + logn

Lim[(nlogn)/(n+logn)] =) Lim 0. As any number time log will grow at a slower rate then adding said number to log. Ex) (10 * log(10)) / (10 + log(10)) = 10 / 11

F(n) = o(g(n))
F(n) = O(g(n))

(2) f(n) = 2^n g(n) = 2^{2n}

Lets say x= 2^n
So rewrite the equations as

F(n) = x and g(n) = x^2

Lim(x/x*x^2) approaches 0 as while the top of the equation will only grow at a rate of x the denominator will be growing faster. Therefore approaching 0.

F(n) = Ω(g(n))
F(n) = ω(g(n))

(3) f(n) = 3^n g(n) = 3^{n+2}

F(n) = O(g(n))
F(n) = Ω(g(n))
F(n) = θ(g(n))

(4) f(n) = $6n^3+7n^2+2$ g(n) = n^3

F(n) = O(g(n))
F(n) = o(g(n))

(5) f(n) = n^{50} g(n) = n!

F(n) = Ω(g(n))
F(n) = ω(g(n))

Question 2 [10 points]

If we know that the running time **T(n)** of some algorithm satisfies the relations **$T(n) = O(n^4 \log n)$** and **$T(n) = \omega(n^2 \log^2 n)$,** which of the following functions can **T(n)** possibly be? Circle **<u>all</u>** that apply.

$n^2 \log n$ n^3 $n^{2.2}$ $n^2 \log^2 n$ $n^4 \log n$

Question 3 [10 points]

Given an algorithm with asymptotic complexity **θ**(n^4), if the running time = 3s when n=100, what is the expected running time when n=200?

100^4 = 100000000 operations total

100000000 / 3 = 33333333.33 operations per second

200^4 = 1600000000 operations
1600000000/ 33333333.33 = 48 seconds total runtime

Question 4 [10 points]

Compute the **<u>largest input size</u>** that each of the following algorithms can solve in **one second**, assuming that they are run on a machine that executes **400 million** operations per second. Compute the best possible numerical value using the given information.
Alg X with an asymptotic complexity of **$\theta(2^n)$**

Log_2(400 million) = 28.58 input size is the maximum

Alg Y with an asymptotic complexity of **$\theta(\sqrt{n})$**

(400 million)^2 = 1.6*10^17 input size is the maximum

Question 5 [40 points]

Find the asymptotic complexity for each of the following algorithms. Be as accurate as possible. Assume that the input size is ***n*** and that ***op*** is a constant-time operation (or possibly sequence of operations). **Show your analysis for full credit.**

(1)

```
for (i=1; i<=n; i*=3).  Θ(logn)
  for (j=0; j<n/5; j+=2).  Θ(n)
    for (k=n; k>=1; k--).  Θ(n)
      op
```

$T(n) = \log n * n * n = n^2 \log n$

(2)

```
for (i=1; i<n; i++).  Θ(n)
  for (j=i; j<i+5; j++).  Θ(5) constant time
      op
```

T(n)= n * 5 = n (as n is the highest order)

(3)

```
for (i=1; i<n; i+=3)    Θ(n)
{
   for (j=4; j<10; j++).  Θ(6) constant time
     op

   for (k=4; k<n/2; k++).  Θ(n)
     op
}
```

T(n) = n(6 + n) = n * n = n^2

(4)

```
for (i=1; i<n; i+=3)    Θ(n)
{
    for (j=1; j<n; j*=2).  Θ(logn)
        op

    x = n*n;
    while(x > 1)     Θ(n^2)
        {op;  x--;}
}
```

T(n) = n (logn + n^2) = nlogn + n^3 = n^3 (as we only care about the highest order)