Week 2 Practical

CP2410: Algorithm and Data Structure

Dang Nhat Quang

13850253

**Task 1:**

Sort from least to greatest

2^10, 2 log n , 3n+100logn, 4n, nlogn, 4nlogn+2n, n^2+10n, n^3, 2^n

**Task 2:**

Algorithm A: 8nlogn

Algorithm B: 2n^2

The graphs describing the behavior of these algorithms start out with A higher (slower) than B, and eventually cross.

After the point where they cross, B is always higher than A.

Therefore, we need to find the point where they cross, that is the value where:

8nlogn = 2n^2.

Applying algebra, we get:

8nlogn = 2n^2

⬄4nlogn=n^2 ⌠

⬄4logn=n

⬄4 = n/logn

Solve for n we have n=16 since 4 =16/log2(16) = 16/4 =4

Thus n0 = 17, since for all n >= 17, A will be faster than B.

**Task 3:**

given that d(n) = O(f(n))

now a\*d(n) = O(a\*f(n)) where a is any constant now, considering Big O notation rule, O(kn) is O(n) where k is constant ⬄ O(a\*f(n)) is also O(f(n)) => Proved

**Task 4:**

Example 1:

there are 2 operations before the loop, the loop runs based off of n, overall runtime is O(n)

Example 2:

2 operations before the loop, the loop runs n / 2 times, Overall runtime = O(n)

Example 3:

2 operations before the loop, inner loop runs 1 + 2 + n times Overall run time is O(n^2)

Example 4:

3 operations before the loop, loop runs n times

Overall run time is O(n)

Example 5:

2 operations before the loop, outer loop runs n times

Overall run time is O(n^3)