Data Structure HW1

Q1.

$\Theta(n ^ 2)$, c1 = 2, c2 = 3, n0 = 1

Q2.

$O((n ^ 2) * (2 ^ n))$, c = 110, n0 = 10

Q3

(1)

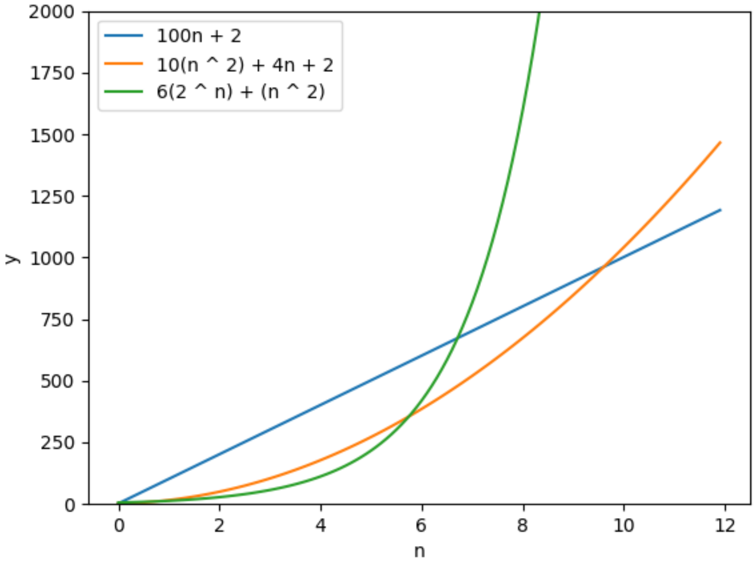
Time Complexity: $\Theta(n ^ 3)$
`i` will iterate from 0 to n - 1, it takes `n` times.
`j` will iterate from 0 to n - 1, it takes `n ^ 2` times.
`k` will iterate from 0 to n - 1, it takes `n ^ 3` times.
Then, `c[i][k] = a[i][j] * b[j][k] + c[I][k]` takes `n ^ 3` times.
In total, it takes `n ^ 3` times, for c1 = 1, c2 = 2 , n0 = 1, its time complexity is $\Theta(n ^ 3)$.

(2)

Time Complexity: $\Theta(n ^ 3)$
`i` will iterate form 0 to n - 1, it takes n times.
`j` will iterate from i to n - 1, it takes `(n - 1) + (n - 2) + ... + 2 + 1 = (n - 1) * (n / 2)`.
`k` will iterate from j to n - 1, it takes `((n - 1) * ((n - 1) + 1) / 2) + ((n - 2) * ((n - 2) + 1) / 2) ... + (1 * (1 + 1) / 2) = ((n ^ 3) - 3(n ^ 2) + 5n) / 6`
Then, `c[i][k] = a[i][j] * b[j][k] + c[I][k]` takes `((n ^ 3) - 3(n ^ 2) + 5n) / 6` times.
In total, it takes `((n ^ 3) - 3(n ^ 2) + 5n) / 6` times, for c1 = 100, c2 = 200 , n0 = 10, its time complexity is $\Theta(n ^ 3)$.

Q4

(1)



(2)

1. `6(2 ^ n) + (n ^ 2) > 10(n ^ 2) + 4n + 2 > 100n + 2`, when `n >= 5`.
2. `Θ(f1) = Θ(n )`, `Θ(f2) = Θ(n ^ 2)`, `Θ(f3) = Θ((2 ^ n) * (n ^ 2))`

(3)

Assume `f(x)` is polynomial, and it time complexity is `O(f)`.
Assume `a > 0`.
Due to `f(x)` time complexity is `O(f)`, then `f(x)= <= c * O(f)`.
There is another polynomial `af(x)`, and `af(x) <= a * c * O(f)`.
So, polynomial `af(x)` time complexity is also `O(f)`.
Thus, we usually ignore coefficient.

(4)

When n is large, low-order terms' effection will smaller. Take `n^2 + n` for instance.

n	1	10	100	1000	10000
n	1	10	100	1000	10000
n^2	1	100	10000	1000000	100000000
n^2+n	2	110	100	1001000	100010000

We find `n^2` is close to `n^2+n` . Relatively, n becomes less important. So, we denote `O(n^2)` instead `O(n^2 + n).`