Assignment 1 Jiajie Xu 7881937

Q1

ⓐ F

$2^{n+a} = 2^n \cdot 2^a > 2^n$ for a positive constant

$2^n > n^2$ for $\forall n > 4$

$\therefore 2^{n+a}$ is $O(2^n)$

ⓑ F

$2^{n+a} = 2^n \cdot 2^n \cdot 2^a > 2^n \quad \forall a \geq 0$

$a^n > n^2 \quad \forall n > 4$

$\therefore 2^{2n+a}$ is $O(2^n)$

ⓒ T

$\sum_{k=1}^{n} k^3 = 1^3 + 2^3 + 3^3 + \cdots + n^3 \leq n \cdot n^3 = n^4 \quad \forall n \geq 1$

$\sum_{k=1}^{n} k^3 \geq n^3 \quad \forall n \geq 1 \quad \sum_{k=1}^{n} k^3 = \Omega(n^3)$

$\therefore \sum_{k=1}^{n} k^3 = O(n^4)$

ⓓ F see ⓒ

ⓔ T

$(2n+8)\log(n^{10}) \leq (2n+8n)\log(n^{10}) \quad \forall n \geq 1$

$= 10(2n+8n)\log(n)$

$= 100 n\log(n)$

$c = 100 \quad g(n) = n\log(n)$

(f) T

$f(n) = (2n+8)\log(n^{14}) \geq n\log(n)$ for $\forall n \geq 1$

$fn = \Omega(n\log(n))$

$\log(n^{14})$ is $O(n\log(n))$

$\therefore \quad f(n) = O(n\log(n))$

Q2

(a) worst case: $\log_3 N$ comparisons

(b)
$T(N) = C + T(N/3)$ ①

$T(N/3) = C + T(N/9)$ ②

$T(N) = T(N/9) + 2C$ ③

$T(N) = T(N/3) + iC$ ④

$T(N/3i) = T(1)$

$N/3i = 1$

$N = 3i$

$T(N) = T(N/N) + iC$

$T(N) = T(1) + iC$

$T(N) = T(1) + C(\log_3 N)$

$T(N) = k + C(\log_3 N) \leq \log_2 N$

$\therefore O(\log_2 N)$

Q3

(a) This algorithms when element sorted find the max length of the subarray

(b) Best $O(n)$
Worst $O(n^2)$

(c) Best: ascending
Worst: descending

Q4 a is the array with n elements

```
for (int i=0; i<n-1; i++)
{
    for (int j = i+1; j<n-1; j++)
    {
        if (a[i]==a[j])
        { return true; }
    }
}
return false
```

Worst: $O(n^2)$