Name: CHAN King Yeung

SID: 1155119394

CSCI2100 Assignment 1

### Question 1

Total frequency:  $3n^2 + 8n + 5$ 

Time complexity:  $O(n^2)$ 

### Question 2

a) By definition, we have  $g_i(n) \le c_i \cdot f_i(n)$  for any  $n \ge n_i$ 

The product of  $g_1(n)$  and  $g_2(n)$  resulting an inequality as follow

$$g_1(n)g_2(n) \le c_1c_2 \cdot f_1(n)f_2(n)$$
 for any  $n > \max(n_1, n_2)$ 

Then we have  $g_1(n)g_2(n) = O(f_1(n)f_2(n))$ 

b) 
$$g(n) = (n^2 + \sqrt{n}) \cdot (n + \log(n)) = O(n^2 \cdot n) = O(n^3)$$

c) 
$$g(n) = (n^3 + 3n^2 + 5) \cdot (n^2 + n^4) = \Theta(n^3 \cdot n^4) = \Theta(n^7)$$

d) By definition, we have  $c_1 \cdot f(n) \le g(n) \le c_2 \cdot f(n)$  for any n

The maximum of f(n) and g(n) resulting an inequality as follow

$$c_1(f(n) + g(n)) \le \max(f(n), g(n)) \le c_2(f(n) + g(n))$$

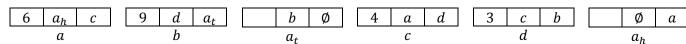
Given both f(n) and g(n) are nonnegative function, we have

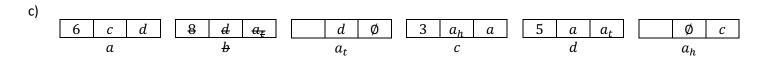
$$max(f(n),g(n)) \le 1 \cdot (f(n)+g(n))$$
 and  $max(f(n),g(n)) \ge \frac{1}{2} \cdot (f(n)+g(n))$ 

Thus,  $c_1=\frac{1}{2}$  and  $c_2=1$  holds the inequality for any n

# Question 3

a) 6





#### Question 4

- a) We have that a=4, b=4,  $\lambda=0$  Since  $log_4(4)>0$ , we have that  $g(n)=O\bigl(n^{log_b(a)}\bigr)=O(n)$
- b) We have that a=2, b=4,  $\lambda=\frac{1}{2}$ Since  $\log_4(2)=\frac{1}{2}$ , we have that  $g(n)=O\left(n^\lambda\cdot\log(n)\right)=O\left(\sqrt{n}\log(n)\right)$
- c) We have that a=2, b=4,  $\lambda=2$ Since  $log_4(2)<2$ , we have that  $g(n)=O(n^{\lambda})=O(n^2)$

#### Question 5

 $f_1(n) = 2^{2^{100000}}$  runs in a constant time, thus,  $f_1(n) = O(1)$ 

 $f_2(n) = 2^{10000n}$  grows in  $2^n$ , thus,  $f_2(n) = O(2^n)$ 

$$f_3(n)={n\choose 2}$$
 rewrite as  $\frac{n(n-1)}{2}$ , thus,  $f_3(n)=O(n^2)$   $f_4(n)=n\sqrt{n}$  known as  $n^{\frac{3}{2}}$ , thus,  $f_4(n)=O\left(n^{\frac{3}{2}}\right)$ 

The growth rate, therefore, is  $f_2(n) > f_4(n) > f_3(n) > f_1(n)$ 

## Question 6

- a) Suppose  $T(n) \le c \cdot n^2$  holds for  $n \le k-1$ For n = k, we have  $T(k) \le T(k-1) + k \le c \cdot (k-1)^2 + k = c \cdot k^2 - 2c \cdot k + c + k$ To make  $T(k) \le c \cdot k^2$ , c must statisfy that  $-2c \cdot k + c + k \le 0 \implies c \ge 1$ By induction, for any  $k \ge 1$ , we obtain  $T(n) \le 1 \cdot n^2$ , therefore,  $T(n) = O(n^2)$
- b) We have that a = 1, b = 2,  $\lambda = 0$ Since  $log_2(1) = 0$ , we have that  $T(n) = O(n^{\lambda} \cdot log(n)) = O(log(n))$

#### Question 7

By the definition of Big-Omega, we obtain as follow

$$g(n) = \Omega\left(n^{\log_b(a)} + \sum_{i=0}^{y} \left(\frac{a}{b^{\lambda}}\right)^i n^{\lambda}\right)$$
, where  $y = \log_b(n) - 1$ 

- a) Given  $log_b(a) < \lambda$ , we have  $a < b^\lambda$  and  $g(n) = \Omega \left( n^{log_b(a)} + c_0 \cdot n^\lambda \right)$  for some constant  $c_0$  $n^{\lambda}$  grow faster than  $n^{\log_b(a)}$ , thus by the sum property we obtain  $g(n) = \Omega(n^{\lambda})$
- b) Given  $log_b(a)=\lambda$ , we have  $a=b^\lambda$  and  $g(n)=\Omega \big(n^{log_b(a)}+n^\lambda \cdot log_b(n)\big)$  $n^{\lambda} \cdot log_b(n)$  grow faster than  $n^{log_b(a)}$ , thus by the sum property we obtain  $g(n) = \Omega(n^{\lambda} \cdot log(n))$
- c) Given  $log_b(a) > \lambda$ , we have  $a > b^{\lambda}$  and  $g(n) = \Omega(n^{log_b(a)} + c_0 \cdot n^{\lambda})$  for some constant  $c_0$  $n^{\log_b(a)}$  grow faster than  $n^{\lambda}$ , thus by the sum property we obtain  $g(n) = \Omega(n^{\log_b(a)})$