HOMEWORK #1 INTRODUCTION TO ALGORITHM

Answers of the Questions HW#1

1)

a) $\log_2 n^2 + 1 \in O(n)$

$\rightarrow f(n) = \log_2 n^2 + 1 \quad g(n) = n$

rule 1) if $\lim_{n\to\infty} f(n)/g(n) = 0$ then $f(n) \in O(g(n))$

$\rightarrow$ we look that limit is zero or not.

$$\rightarrow \lim_{n\to\infty} \frac{\log_2 n^2 + 1}{n} \Rightarrow \lim_{n\to\infty} \frac{\frac{\ln(n^2)}{\ln(2)} + 1}{n}$$

$\Rightarrow$ If we use L'hospital rule.

$$\Rightarrow \lim_{n\to\infty} \frac{\frac{d}{dn}\left(\frac{\ln(n^2)}{\ln(2)} + 1\right)}{\frac{d}{dn}(n)} \Rightarrow \lim_{n\to\infty} \frac{2}{n \ln(2)}$$

$\Rightarrow \frac{\text{number}}{\infty}$ is equal 0 So $\lim_{n\to\infty} \frac{\log_2 n^2 + 1}{n} = 0$

So with respect to rule 1) $\Rightarrow$ $\boxed{\log_2 n^2 + 1 \in O(n) \text{ is True}}$

b) $\sqrt{n\cdot(n+1)} \in \Omega(n)$

$\rightarrow$ We check $\sqrt{n^2+n} \geq f(n)$ is true or not for Ω notation.

$\sqrt{n^2+n} \cong n + \sqrt{n}$ So $n + \sqrt{n} > n \Rightarrow \sqrt{n^2+n} \geq n$

$\rightarrow$ So $\sqrt{n\cdot(n+1)} \geq n$ then $\sqrt{n\cdot(n+1)} \geq n$

$\rightarrow$ $\boxed{\sqrt{n\cdot(n+1)} \in \Omega(n) \text{ is True}}$

c) $n^{n-1} \in \Theta(n^n)$

→ we look both big O notation and Ω notation
becouse of Θ.

→ First we look $n^{n-1} \in O(n^n)$ is true or not

→ By using limit approach and using rule 1)

$\lim_{n \to \infty} \frac{n^{n-1}}{n^n} \Rightarrow \frac{n^{n-1}}{n \cdot n^{n-1}} \Rightarrow \frac{1}{n} = \frac{1}{\infty} = 0$ So

$n^{n-1} \in O(n^n)$

→ Now we look $n^{n-1} \in \Omega(n^n)$ true or not

→ $n^{n-1} \geq n^n$ is not true

So $n^{n-1} \notin \Omega(n^n)$

Therefore $n^{n-1} \in \Theta(n^n)$ is NOT true.

d) $O(2^n + n^3) \subset O(4^n)$

→ $O(2^n + n^2) \subset O(2^{2n})$

$O(2^n + n^2) \in O(2^{2n})$ due to $O(2^n) \subset O(2^{2n})$

So this $O(2^n + n^3) \subset O(4^n)$ is true.

e) $O(2\log_3 \sqrt[3]{n}) \subset O(3\log_2 n^2)$

→ $O(2\log_3 \sqrt[3]{n}) \in O(\log_3 \sqrt[3]{n}) \subset \log_2 n^2$

$\frac{2}{3} \frac{1}{\log_n 3} < \frac{3}{2} \frac{1}{\log_n 2}$

So $O(2\log_3 \sqrt[3]{n}) \subset O(3\log_2 n^2)$ is true

f) $\log_2\sqrt{n}$ and $(\log_2 n)^2$ have the same notation or not.

$\rightarrow O(\log_2\sqrt{n}) = O(\frac{1}{2}\log_2 n)$

$\rightarrow O(\log_2^2 n)$

$O(\log_2^2 n) \neq O(\log_2 n)$ are not the same notation

$$\boxed{\text{So this proposition is not true.}}$$

2) Order $n^2, n^3, n^2\log n, \sqrt{n}, \log n, 10^n, 2^n, 8^{\log n}$ (log base 2)

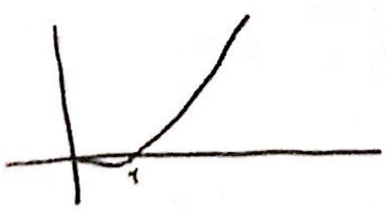
$\rightarrow n^2$ growth rate

$n=100 \Rightarrow 10000$
$n=1000 \Rightarrow 1000000$

$\rightarrow n^3$ growth rate

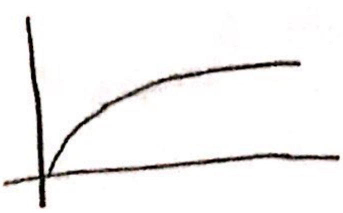
$n=100 \Rightarrow 1000000$
$n=1000 \Rightarrow 10^9$

$\rightarrow n^2\log n$



$n=100 \Rightarrow 66400$
$n=1000 \Rightarrow 9960000$

$\rightarrow \sqrt{n}$



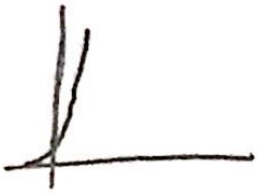
$n=100 \Rightarrow 10$
$n=1000 \Rightarrow 31{,}62$

$\longrightarrow \log n$

$n=100 \Rightarrow 2$

$n=1000 \Rightarrow 3$

$\longrightarrow 10^n$



$n=100 \Rightarrow 10^{100}$

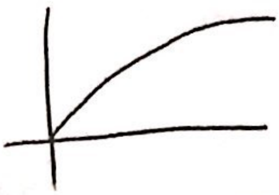
$n=1000 \Rightarrow 10^{1000}$

$\longrightarrow 2^n$

$n=100 \Rightarrow 2^{100}$

$n=1000 \Rightarrow 2^{1000}$

$\longrightarrow 8^{\log n}$



$n=100 \cong 992\,013$

$n=1000 \cong 988043966$

$$\boxed{\text{So} \quad \log n < \sqrt{n} < n^2 < n^2 \log n < 8^{\log n} < n^3 < 2^n < 10^n}$$

3)

a)

```
void f(int my-array[]){
    for (int i=0; i<sizeofArray; i++){
        if(my_array[i]<first_element){
            second_element=first_element; -> O(1)
            first_element=my_array[i]; -> O(1)
        }
        else if(my-array[i]<second-element){
            if(my_array[i]!=first-element){
                Second-element=my-array[i];
                    -> O(1)
            }
        }
    }
}
```

O(1)

O(1)

O(sizeofArray)

sizeofArray=n

$\boxed{O(n)}$

I showed complexity of the lines on the function.
Therefore the complexity of a) $\longrightarrow O(\text{sizeofArray}) \to O(n)$
Because i increases one-by-one
$O(1) \cdot O(1) \cdot O(1)$ is equal $\boxed{O(n)}$

It is also $\Omega(n)$
So the complexity of this function f is $\boxed{\Theta(n)}$

b)
```
void f(int n){
        int count=0
        for(int i=2; i<=n; i++){
                if(i%2==0){
                        count++;
                }
                else{
                        i=(i-1)*i
                }
        }
}
```

In the else part i becomes $i * i-1 = i^2 - i$
then i increases one so i becomes $i = i^2 - i + 1$
but to find complexity I ignore the small side $-i+1$ so I look the i^2 part to find complexity.
i^2, $i=3$ then, $i = 3, 3^2, (3^2)^2, (3^2)^4, (3^2)^8, \ldots$

$$3^{2^k} = n \qquad 2^k = \log_3 n \qquad k = \log_2(\log_3 n)$$

So complexity of this function is $\boxed{O(\log(\log n))}$

4)

a) $\sum_{i=1}^{n} i^2 \log_2 i \Rightarrow \underbrace{1^2 \log_2 1}_{0} + \sum_{i=2}^{n} i^2 \log i \Rightarrow H(n) = \sum_{i=2}^{n} i^2 \log i$

$$\Rightarrow \int_1^n i^2 \log i \, di \leq f(n) \leq \int_1^{n+1} i^2 \, di$$

$$\Rightarrow \int \frac{i^2 \ln(i)}{\ln(2)} di \Rightarrow \frac{1}{\ln(2)} \int i^2 \ln(i) \, di$$

$$\Rightarrow \int i^2 \ln(i) \, di = \frac{i^3 \ln(i)}{3} - \int \frac{i^2}{3} di$$

$$\Rightarrow \int \frac{i^2}{3} di = \frac{i^3}{9}$$

$$\Rightarrow \frac{i^3 \ln(i)}{3 \ln(2)} - \frac{i^3}{9 \ln(2)}$$

$$\Rightarrow \frac{i^3 (3 \ln(i) - 1)}{9 \ln(2)}$$

$$\left. \frac{i^3 (3 \ln(i) - 1)}{9 \ln(2)} \right|_1^n \leq H(n) \leq \overbrace{\left. \frac{i^3 (3 \ln(i) - 1)}{9 \ln(2)} \right|_1^{n+1}}^{G(x)}$$

$$\frac{n^3 (3 \ln(n) - 1)}{9 \ln(2)} - \frac{1 \cdot (3 \cdot \overbrace{\ln(1)}^{0} - 1)}{9 \ln(2)} = \frac{n^3 (3 \ln(n) - 1) + 1}{9 \ln(2)}$$

$$G(x) = \frac{(n+1)^3 (3 \ln(n+1) - 1)}{9 \ln(2)} - \frac{1}{9 \ln(2)} = \frac{(n+1)^3 (3 \ln(n+1) - 1)}{9 \ln(2)}$$

$$\frac{n^3 (3 \ln(n) - 1) + 1}{9 \ln(2)} \leq H(n) \leq \frac{(n+1)^3 (3 \ln(n+1) - 1)}{9 \ln(2)}$$

$H(n) \in O(n^3 \log n) \quad H(n) \in \Omega(n^3 \log n)$

$$\boxed{\text{So } \sum_{i=1}^{n} i^2 \log i \in \Theta(n^3 \log n)}$$

b) $\sum_{i=1}^{n} i^3$

$$\rightarrow \int_0^n i^3\,di \leq \underbrace{\sum_{i=1}^{n} i^3}_{H(n)} \leq \int_1^{n+1} i^3\,di$$

$$\frac{i^4}{4}\Big|_0^n \leq H(n) \leq \frac{i^4}{4}\Big|_1^{n+1}$$

$$\frac{i^4}{4}\Big|_0^n = \frac{n^4}{4} \qquad \frac{i^4}{4}\Big|_1^{n+1} = \frac{(n+1)^4-1}{4}$$

$$\frac{n^4}{4} \leq H(n) \leq \frac{(n+1)^4-1}{4}$$

$H(n) \in O(n^4)$ $\quad$ $H(n) \in \Omega(n^4)$

$$\boxed{\text{So } H(n) = \sum_{i=1}^{n} i^3 \in \Theta(n^4)}$$

c) $\sum_{i=1}^{n} \frac{1}{2\sqrt{i}}$

$$\int_1^{n+1} \frac{1}{2} i^{-1/2} \leq H(n) \leq \int_0^n \frac{1}{2} i^{-1/2}$$

$$\frac{\frac{1}{2} i^{1/2}}{\frac{1}{2}}\Big|_1^{n+1} = \sqrt{n+1} - 1 \qquad \frac{\frac{1}{2} i^{-1/2}}{\frac{1}{2}}\Big|_0^n = \sqrt{n}$$

$$\sqrt{n+1} - 1 \leq H(n) \leq \sqrt{n}$$

$H(n) \in O(\sqrt{n})$ $\quad$ $H(n) \in \Omega(\sqrt{n})$

$$\boxed{\text{So } H(n) = \sum_{i=2}^{n} \frac{1}{2\sqrt{i}} \in \Theta(\sqrt{n})}$$

d) $\sum_{i=1}^{n} \frac{1}{i}$

$H(n) = 1 + \sum_{i=2}^{n} \frac{1}{i}$ for upper bound because of undefined $\frac{1}{0}$

$$\int_{1}^{n+1} \frac{1}{i}\, di \le H(n) \le 1 + \int_{1}^{n} \frac{1}{i}\, di$$

$$\ln(i) \Big|_{1}^{n+1} = \ln(n+1) - \overbrace{\ln(1)}^{0} = \ln(n+1)$$

$$\ln(i) \Big|_{1}^{n} = \ln(n) - \overbrace{\ln(1)}^{0} = \ln(n)$$

$$\Rightarrow \ln(n+1) \le H(n) \le \ln(n) + 1$$

$$H(n) \in O(\log n) \quad H(n) \in \Omega(\log n)$$

$$\boxed{\text{So } H(n) = \sum_{i=1}^{n} \frac{1}{i} \in \Theta(\log n)}$$

5) Linear search with repeated elements

The Best Case $O(1)$ because if the searched element in the list, due to all elements are the same, the first element in the list is the searched element. Therefore $L[1]$ is searched element and the complexity of this is $O(1)$ that is the best case.

The Worst Case $O(n)$ because if the searched element is not in the list then complexity is $O(n)$. if $x \neq L[n]$ it should look all elements so the worst complexity is $O(n)$ if the searched element is not in the list.