CSE 321 Homework #1

Muhammed Yasir Fidan
161044056

1)

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

$\log_2 n^2 = 2\log_2 n$

$$\lim_{n\to\infty} \frac{2\log n + 1}{n} \xrightarrow{L'Hospital} \lim_{n\to\infty} \frac{\frac{2}{n\ln 2}}{1} = \frac{2}{\infty} = 0$$

Since our limit result is 0, it means $\log_2 n^2 + 1$ has a smaller order growth rate than n so;

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

its true

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

$$\lim_{n\to\infty} \frac{\sqrt{n(n+1)}}{n} = \frac{\sqrt{n^2+n}}{n}$$ → lets ignore this low term n

$$= \frac{\sqrt{n^2}}{n} = \frac{n}{n} = 1$$

because our limit result is a constant;

$\sqrt{n(n+1)} \in \Theta(n)$

it also mean

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

so its true

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

$n^{n-1} = \frac{n^n}{n}$ $\quad \lim_{n\to\infty} \frac{\frac{n^n}{n}}{n^n} = \lim_{n\to\infty} \frac{1}{n} = \frac{1}{\infty} = 0$

because of limit result is 0, $n^{n-1} \notin \Theta(n^n)$

actualy its $n^{n-1} \in O(n^n)$ so its false

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

let say $f(n) = 2^n + n^3$ and $g(n) = 4^n$

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{2^n + n^3}{4^n} = \underbrace{\lim_{n\to\infty} \frac{1}{2^n}}_{0} + \lim_{n\to\infty} \frac{n^3}{2^{2n}}$$

$$\lim_{n\to\infty} \frac{n^3}{4^n} \xrightarrow{\text{L'Hospital}} \lim_{n\to\infty} \frac{3n^2}{4^n \ln 4} \xrightarrow{\text{L'Hospital}} \lim_{n\to\infty} \frac{6n}{4^n \ln 4 . \ln 4}$$

$$\lim_{n\to\infty} \frac{6n}{4^n \ln 4 \ln 4} \xrightarrow{\text{L'Hospital}} \lim_{n\to\infty} \frac{6}{4^n \ln 4 . \ln 4 . \ln 4} = \frac{6}{\infty} = 0$$

because limit result is 0, $f(n) \in O(g(n))$

$2^n + n^3 \in 4^n$

And from the propor ties of complexities we know;

$f(n) \in O(g(n)) \Longleftrightarrow O(f(n)) \subseteq O(g(n))$

so; $O(2^n + n^3) \subset O(4^n)$ true

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

lets $f(n) = 2\log_3 \sqrt[3]{n}$ $g(n) = 3\log_2 n^2$

$f(n) = 2\log_3 n^{\frac{1}{3}} = \frac{2}{3}\log_3 n$

$g(n) = 3\log_2 n^2 = 6\log_2 n$

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{\frac{2}{3}\log_3 n}{6\log_2 n} = \lim_{n\to\infty} \frac{1}{9} \cdot \frac{\log_3 n}{\log_2 n}$$

Apply L'Hospital $\lim_{n\to\infty} \frac{1}{9} \frac{(\log_3 n)'}{(\log_2 n)'} = \lim_{n\to\infty} \frac{1}{9} \cdot \frac{\ln 2}{\ln 3} = c$

because of we found a constant $2\log_3 \sqrt[3]{n} \in \Theta(3\log_2 n^2)$

it also mean

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

And from the proportie

$f(n) \in O(g(n)) \Longleftrightarrow O(f(n)) \subseteq O(g(n))$

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

f) $\log_2\sqrt{n}$ and $(\log_2 n)^2$ same order?

$$\lim_{n\to\infty} \frac{\log_2\sqrt{n}}{(\log_2 n)^2}$$

let say $n = x^2$

$$\lim_{n\to\infty}\frac{\log_2 x}{(\log_2 x^2)^2} = \lim_{n\to\infty}\frac{\log_2 x}{4(\log_2 x)^2} = \lim_{n\to\infty}\frac{1}{4}\cdot\log_2 x^{-1}$$

$$= \lim_{n\to\infty}\frac{1}{4}\cdot\frac{1}{\log_2\sqrt{n}} = \lim_{n\to\infty}\frac{1}{2}\frac{1}{\log_2 n} = \frac{1}{2}\cdot\lim_{n\to\infty}\frac{1}{\log_2 n} = 0$$

For same order it has to $\log_2\sqrt{n} \in \Theta((\log_2 n)^2)$

but limit result 0 so; $\log_2\sqrt{n} \notin \Theta((\log_2 n)^2)$

because of that they are not same order.

2) Order the following by growth rate and explain

$$\underbrace{n^2, n^3, n^2\log n, \sqrt{n}, \log n}_{\text{Polynomial}}, \underbrace{10^n, 2^n, 8^{\log n}}_{\text{Exponential}}$$

- Lets compare n^2 and n^3

$\lim\limits_{n\to\infty} \frac{n^2}{n^3} = \lim\limits_{n\to\infty} \frac{1}{n} = 0$ so; $n^3 > n^2$ (1)

- Compare $\sqrt{n}$ and n^2

$\lim\limits_{n\to\infty} \frac{\sqrt{n}}{n^2} = \lim\limits_{n\to\infty} \frac{1}{n^{3/4}} = 0$ so; $n^2 > \sqrt{n}$ (2)

from (1) and (2) $n^3 > n^2 > \sqrt{n}$

- Compare $\log n$ and $\sqrt{n}$

$\lim\limits_{n\to\infty} \frac{\log n}{\sqrt{n}} \xrightarrow{\text{L'Hospital}} \lim\limits_{n\to\infty} \frac{\frac{1}{n}}{\frac{1}{2}\cdot\frac{1}{\sqrt{n}}} = \lim\limits_{n\to\infty} \frac{2\sqrt{n}}{n} = \lim\limits_{n\to\infty} \frac{2}{\sqrt{n}} = 0$

so; $\sqrt{n} > \log n$ (3)

from (1), (2) and (3) $n^3 > n^2 > \sqrt{n} > \log n$

- Compare n^2 and $n^2\log n$

$\lim\limits_{n\to\infty} \frac{n^2}{n^2\log n} = 0$ so $n^2\log n > n^2$ (4)

- Compare n^3 and $n^2\log n$

$\lim\limits_{n\to\infty} \frac{n^3}{n^2\log n} = \lim\limits_{n\to\infty} \frac{n}{\log n} \xrightarrow{\text{L'Hospital}} \lim\limits_{n\to\infty} \frac{1}{\frac{1}{n}} = \infty$

so; $n^3 > n^2\log n$ (5)

from (4) and (5) $n^3 > n^2\log n > n^2 > \sqrt{n} > \log n$

We found polynomial ones growth rate, now lets find exponential ones

- Compare 10^n and 2^n

$\lim_{n\to\infty} \frac{10^n}{2^n} = \lim_{n\to\infty} 5^n = \infty$ so; $10^n > 2^n$ (6)

- Compare 2^n and $8^{\log n}$

$\lim_{n\to\infty} \frac{2^n}{8^{\log n}} = \lim_{n\to\infty} \frac{2^n}{2^{3\log n}}$

from the previous ones we know n grow rate bigger than $\log n$
so this limit goes ∞, because of that $2^n > 8^{\log n}$ (7)

so from (6) and (7) exponential ones $\underline{10^n > 2^n > 8^{\log n}}$

- Compare $8^{\log n}$ and n^3

$\lim_{n\to\infty} \frac{8^{\log n}}{n^3} = \lim_{n\to\infty} \frac{n^3}{n^3} = 1$ so their grow rate same

After this with all this equations, we found our result as;

$$\underline{10^n > 2^n > 8^{\log n} = n^3 > n^2\log n > n^2 > \sqrt{n} > \log n}$$

3) What is the time complexity of the following programs? Explain by giving details.

a)

```
              void f(int my_array[]) {
Θ(n)  ————      for(int i=0; i<sizeofArry; i++){
Θ(1)  ————          if(my_array[i]<first_element){
Θ(1)  ————              second_element=first_element;
Θ(1)  ————              first_element = my_array[i];
                    }
Θ(1)  ————          else if(my_array[i]<second_element){
Θ(1)  ————              if(my_array[i] != first_element){
Θ(1)  ————                  second_element = my_array[i];
                        }
                    }
                }
              }
```

In this algorithm, Our for loop time complexity $\Theta(n)$ becuse we don't change sizeofArray in loop and i value increase one by one everytime. So, this loop just traverse every index of given array. and every statement in loop body has constant time complexity every time. Because of that this function time complexity will be $\underline{\Theta(n)}$.

$$\sum_{i=1}^{n} c.1 = c.n \in \Theta(n)$$ for best, worst and average case

Some constant time complexity statements in for loop 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;
        }
    }
}
```

$i = i^2 - i$

Lets ignore this i because its low term

And ignore i++ part because i^2 growth rate much bigger.

Also lets discard $i\%2==0$ part because for even i values it just increment i by 1 but for odd i values, we change i much larger. So, let's think in every iteration we change i to i^2.

<u>i values</u>

$$\left.\begin{array}{l} 2^{2^0} \\ 2^{2^1} \\ 2^{2^2} \\ \vdots \\ 2^{2^N} \end{array}\right\}$$

$n > 2^{2^N} \Rightarrow \log n > 2^N \Rightarrow \log\log n > N$

We found an upper bound

$O(\log\log n)$

4-) Find the complexity classes of the following functions using the integration method.

a) $\sum_{i=1}^{n} i^2 \log i$ $\quad f(n) = \sum_{i=1}^{n} i^2 \log i \quad g(n) = i^2 \log i$

$f(n) = 1.g(1) + 1.g(2) + 1.g(3) \ldots + 1.g(n)$

The entire area under the curve, $f(n) \leqslant \int_1^{n+1} g(n)\,dn$

$$\int_0^n g(n)\,dn \leqslant f(n) \leqslant \int_1^{n+1} g(n)\,dn \quad \Bigg\}$$ This is valid because g is a non-decreasing function.

$$\int_0^n x^2 \log x\,dx \leqslant f(n) \leqslant \int_1^{n+1} x^2 \log x\,dx$$

$\log x = u \qquad dv = x^2 dx \qquad \log x . \frac{x^3}{3} - \int \frac{x^3}{3} \frac{1}{x} dx$

$\frac{1}{x} dx = du \qquad V = \frac{x^3}{3}$

$du = \frac{dx}{x} \qquad = \log x . \frac{x^3}{3} - \frac{1}{9} . x^3 = \frac{x^3}{3}(\log x - \frac{1}{3})$

$$\frac{x^3}{3}(\log x - \frac{1}{3}) \Big|_0^n = \frac{n^3}{3}(\log n - \frac{1}{3})$$

$$\frac{x^3}{3}(\log x - \frac{1}{3}) \Big|_1^{n+1} = \frac{(n+1)^3}{3} . (\log(n+1) - \frac{1}{3})$$

$$\frac{n^3}{3}(\log n - \frac{1}{3}) \leqslant f(n) \leqslant \frac{(n+1)^3}{3} . (\log(n+1) - \frac{1}{3})$$

After ignore constants and low terms

$n^3 \log n \leqslant f(n) \leqslant n^3 \log n \quad$ so; $\underline{f(n) \in \Theta(n^3 \log n)}$

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

$$f(n) = \sum_{i=1}^{n} i^3 \qquad f(n) = 1.g(1) + 1.g(2) + 1.g(3) + \dots + 1.g(n)$$

$$f(n) \leq \int_1^{n+1} g(x)\,dx$$

$$\int_0^n g(x)\,dx \leq f(n) \leq \int_1^{n+1} g(x)\,dx$$

} I can say that because f(n) is a non-decreasing function

$$\int_0^n x^3 dx = \frac{x^4}{4}\Big|_0^n = \frac{n^4}{4}$$

$$\int_1^{n+1} x^3 dx = \frac{x^4}{4}\Big|_1^{n+1} = \frac{(n+1)^4}{4} - \frac{1}{4}$$

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

After ignore constants and low terms this both n^4

so; $f(n) \in \Theta(n^4)$

c) $\sum_{i=1}^{n} \frac{1}{2\sqrt{i}}$ This is a non-increasing function

$$\int_1^{n+1} g(x)\,dx \leq f(n) \leq \int_0^{n} g(x)\,dx$$ } for a non-increase function

$$\int \frac{1}{2\sqrt{x}}\,dx = \sqrt{x}$$

$$\Big|_1^{n+1} \sqrt{x} \leq f(n) \leq \Big|_0^{n} \sqrt{x}$$

$\sqrt{n+1} - 1 \leq f(n) \leq \sqrt{n}$ $\quad f(n) \in \Theta(\sqrt{n})$

d) $\sum_{i=1}^{n} \frac{1}{i}$ This is a non-increasing function

$$\int_1^{n+1} g(x)\,dx \leq f(n) \leq \int_0^{n} g(x)\,dx$$ } Because its non-increasing function

$$\int \frac{1}{x}\,dx = \ln x$$

$$\ln x \Big|_1^{n+1} \leq f(n) \leq \ln x \Big|_0^{n} \longrightarrow \ln(n+1) \leq f(n) \leq \underbrace{\ln n - \ln(0)}_{\infty}$$

We have a lower bound but we couldn't find a upper bound because $O(\infty)$ is meaningless bound.

$$f(n) = \sum_{i=1}^{n} \frac{1}{i} = 1 + \sum_{i=2}^{n} \frac{1}{i}$$

now upper bound $1+\int_1^n g(x)\,dx$

$$1+\ln x\Big|_1^n = 1+\ln(n)$$

$$\ln|n+1| \leq f(n) \leq \ln|n| + 1$$

from this $f(n) \in \Theta(\log n)$

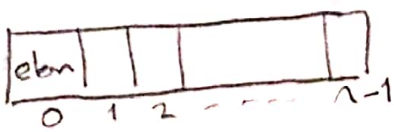
5-) Find the best case and worst case complexities of linear search with repeated elements, that is, the elements in the list need not be distinct. Show your anlysis

As we see in class, linear Search;

```
LinearSearch (list, elem)
    for i to list-length:
        if (list[i] == elem)
            return i
return ∅
```

Comprison is basic operation

- Its the best case If the first element is the element that we are looking for. It just do one comparison and return. Because of that best case $\Theta(1)$



$B(n) = 1 \in \Theta(1)$

- For worst case element that we are searching only my be in the last position or this element my not be in the list at all. for both of this cases its do n comparision. So worst case time complexity $\Theta(n)$

$$W(n) = \sum_{i=1}^{n} 1 = n \in \Theta(n)$$