CSE 222   Midterm Examination   Muhammed Bedir ULUÇAY

I hereby pledge on my honor that I will strictly adhere to academic integrity codes and the work done on this examination is solely my own and I will not receive /give any help from/to anybody or source during this examination.

1-) for ArrayList.   (Assuming i starts 1) (Otherwise it will be infinite loop) = $O(\infty)$

a) sum = 0 => $\Theta(1)$

```
for(int i=1 ; i<myList.size(); i*=2)      => Θ(log n)
    sum += myList.get(i);                  => Θ(1)
```

This method T(n) for ArrayList is constant

$$T(n) = \underbrace{\Theta(1)}_{1} + \left(\underbrace{\Theta(\log(n))}_{2} * \underbrace{\Theta(1)}_{3}\right)$$

- We can ignore 1 according to the following rule

$$T(n) = max(T_1(n), T_2(n))$$

- in for loop i is ingreasing $i = i*2 \Rightarrow (\log n)$

$$\boxed{T(n) = \Theta(\log(n))}$$

b) for LinkedList

sum=0 => $\Theta(1)$ ignore

```
for(int i = 1; i < myList.size(); i*=2)   => Θ(log n)
    sum += myList.get(i);
```

This is working $\Theta(n)$ time for LinkedList

$$T_n = \Theta_1 + (\Theta(\log n) * \Theta(n)) \qquad \boxed{T(n) = \Theta(n \log n)}$$

2-) `int foo( AL l) {`

`if(0= l.size()) return 0;` $\Rightarrow \Theta(1)$

`int sum = 0;` $\Rightarrow \Theta(1)$

`int x = l.get(l.size()-1);` $\Rightarrow \Theta(1)$ contsant for Arraylist

`for (int i=0; i< l.size(); ++i)` $\Rightarrow \underline{\Theta(n)}$

`sum += x%i` $\Rightarrow \Theta(1)$

`l.remove(l.size()-1);` $\Rightarrow \Theta(1)$ because deleting last element not shifting array for each time

`sum += foo(l);` $\Rightarrow \Theta(n)$ → it will call l.size() times $\Theta(n)$

`return sum;`

`}`

$$\frac{n.(n+1)}{2} = n^2 \text{ (ignore const)}$$

$$T(n) = T(n-1) + \Theta(n)$$
$$T(n-1) = T(n-2) + \Theta(n-1)$$
$$\vdots \qquad \vdots \qquad \vdots$$

$$T(n) = \Theta(n) * \Theta(n)$$
$$\boxed{T(n) = \Theta(n^2)}$$

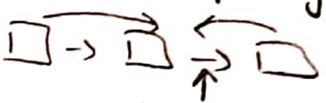
3-)

a) $O(n \log n)$ ✓ → Because it will run max lenght of list it is $O(n)$ we can accept $O(n \log n)$ because $n \log n > n$

$\Theta(n)$ ✓ → for this function it is more detailed answer it would be this. because it will process L.size() time in while loop

$\Omega(n^2)$ ✗ → We cannot accept this because omega notation must be lowes than $\Theta(n)$ notation $n^2 < n$ this is wrong

b) $O(n \log n)$ ✓ → it will complexity for the take an index



max it will be $O(\frac{n}{2})$

so $\underline{n \log n > \frac{n}{2}}$ we can accept it because of O notation

$\Theta(n)$ ✓ → This is also correct $\Theta(\frac{n}{2}) = \Theta(n)$ we can ignore constant

$\Omega(n^2)$ ✗ → We can't accept this because $n^2 < n/2$ is false for omega notation