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.

CSE 222
Midterm Examination
Süleyman Galbal
1801042656
~~55~~.

1)

Every time in loop the counter i will be multiply by 2, if we say n to executed times of loop, s to the size of myList

$n = 2^s$ so $\log_2 n = \log_2 2^s$ so $s = \log_2 n$

a) In arraylist get directly finds that value with $\Theta(1)$ because we can add the index to memory adress and find the value directly.

So $T(n) = \log_2 n + \Theta(1)$

constant time so doesn't effect $\Theta(n) = \log_2 n$

b) In linkedlist get method finds value in best case $O(1)$ (the head) and $O(n)$ in worst case (End of linked list so it has to traverse). So avarage case is $\Theta(\frac{n}{2})$ which $\frac{1}{2}$ constant so $\Theta(n)$.

When we combine $T(n) = (\log_2 n) \cdot n = n \log n$

2)

```
int foo (Arraylist l, int sum, int i){
    if(l.size == 0){
        l.remove(l.size() - 1);
        sum += foo(l, sum, 0);
        return sum;
    }
    else{
        sum += l.get(l.size()-1); // x value
        return foo(l, sum, ++i);
    }
}
```

CSE 222
Midterm Examination
Silayra Gaibat
1801042656

3)

a) for a, best case is $\Omega(1)$ because there is just one node

$\Theta(n \log n)$ X
$\Theta(n)$ X
$\Omega(n^2)$ ✓

worst case is $\Omega(n^2)$ because the String is immutable every time a new String creating with the (size-1). While loop goes n time and toString goes too. So $n \cdot n = n^2$ in worst case because best case doesn't have nlogn.

nlogn wrong