Week 1 Quiz

Q1.

What's the time complexity to add a node to the head of a linked list?

A. `O(1)`

B. `O(logn)`

C. `O(n)`

D. `O(n^2)`

Q2.

What is the big-O running time of the following function?

```
python
def mystery(lst):
    sum = 0
    for i in range(len(lst) - 1):
        if i % 2 == 0: # i is even
            total += 5 * lst[i] - 1

    for i in range(len(lst) / 2):
        for j in range(500 * len(lst)):
            if i < j + 1:
                total -= j

    return total
```

A. O(1)

B. O(n)

C. O(nlogn)

D. O(n^2)

Q3.

The efficiency of summing all of the keys in a "regular" (non-search) binary tree is `O(n)`, since the algorithm needs to visit every node. What is the efficiency of summing all of the keys of a binary search tree?

A. `O(1)`

B. `O(logn)`

C. `O(n)`

D. `O(nlogn)`