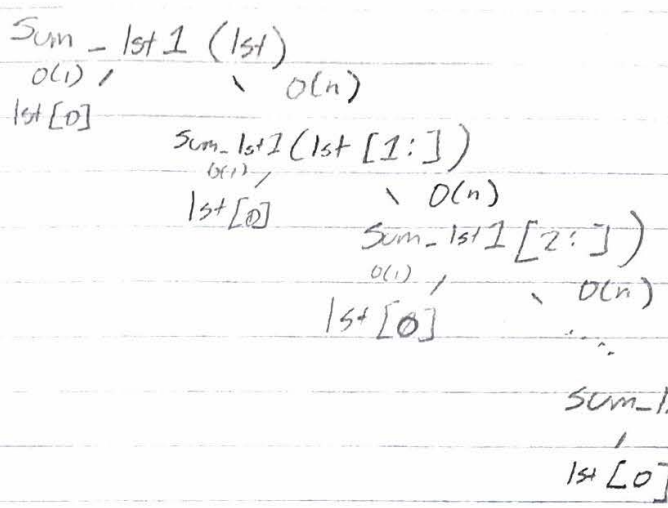
tg1632_hw3.pdf

Question 1:

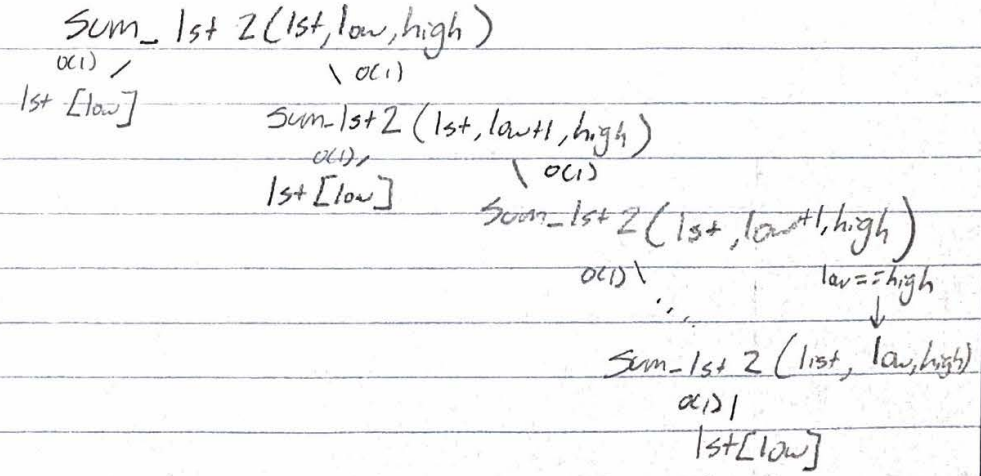
sum_lst1



Asymptotic Running Time: Each recursive call an element in the list is accessed which is O(1). You do this n times, hence O(n). Also you call the recursive function with a spliced version of the original list. Splicing is O(n) and this is done n times, hence it is $O(n^2)$.

Final Running Time: $\boxed{O(n^2)}$

sum_lst2

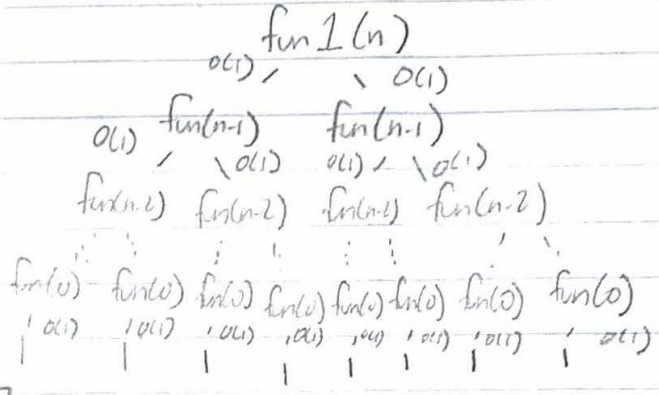


Asymptotic Running Analysis: Each recursive call returns an element in the list which is O(1). This is done n times (n being the len of lst), hence making it O(n). The recursive call is O(1) n times hence $\boxed{O(n)}$.

☆ $\boxed{\text{Sum\_lst2}}$ is asymptotically faster.

Question 2:

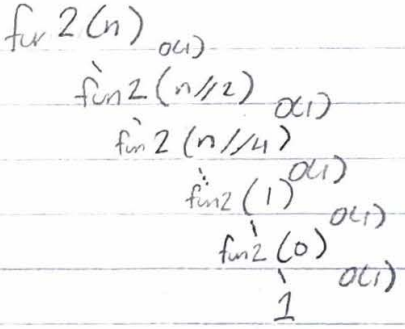
(a) (i)



All recursive calls O(1)

(ii) $\boxed{O(2^n)}$

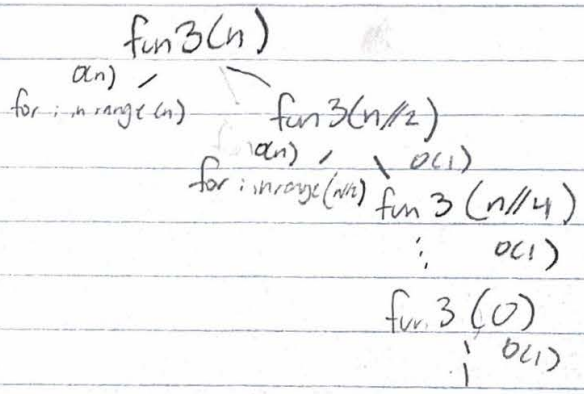
(b) (i)



All recursive calls O(1)

(ii) $\boxed{O(\log n)}$

(c)



(ii) O(1) log n times and O(n) · n times = $\boxed{O(n^2 \log n)}$