

3. (a)

def foo1 (lst):

 $i = \text{len}(lst) \times 2 - \theta(1)$
for j in range($\text{len}(lst) - 1$):
 $lst[j], lst[j+1] = lst[j+1], lst[j] - \theta(1)$

 if $\frac{j+i}{2} \leq j$: $j - \theta(1)$
 break
 $i // 2 - \theta(1)$] $\theta(n)$

$$\therefore T(n) = \theta(n)$$

(b) def foo2 (lst):

 $n = \text{len}(lst) - \theta(1)$ for i in range($n - 1$): $j = i + 1 - \theta(1)$ while $j > 1$: $lst[j] = 3 \times j - i$ if $lst[j] > lst[j-1] / 2$:else: $j -= 1 - \theta(1)$ $j -= 3 - \theta(1)$
 $\theta(n)$] $\theta(n^2)$

$$\therefore T(n) = \theta(n^2)$$

4. ~~Q~~ list = [1, 2, 3, 4, 5, 6, 7, 8] low = 0 high = 7

(a) sum_list1's output and return:

sum_list1, low = 0, high = 7

sum_list1, low = 1, high = 7

sum_list1, low = 2, high = 7

sum_list1, low = 3, high = 7

sum_list1, low = 4, high = 7

sum_list1, low = 5, high = 7

sum_list1, low = 6, high = 7

sum_list1, low = 7, high = 7

36

~~Q~~ sum_list2's output and return:

sum_list2, low = 0, high = 7

sum_list2, low = 0, high = 3

sum_list2, low = 0, high = 1

sum_list2, low = 0, high = 0

sum_list2, low = 1, high = 1

sum_list2, low = 2, high = 3

sum_list2, low = 2, high = 2

sum_list2, low = 3, high = 3

sum_list2, low = 4, high = 7

sum_list2, low = 4, high = 5

sum_list2, low = 4, high = 4

sum_list2, low = 5, high = 5

sum_list2, low = 6, high = 7

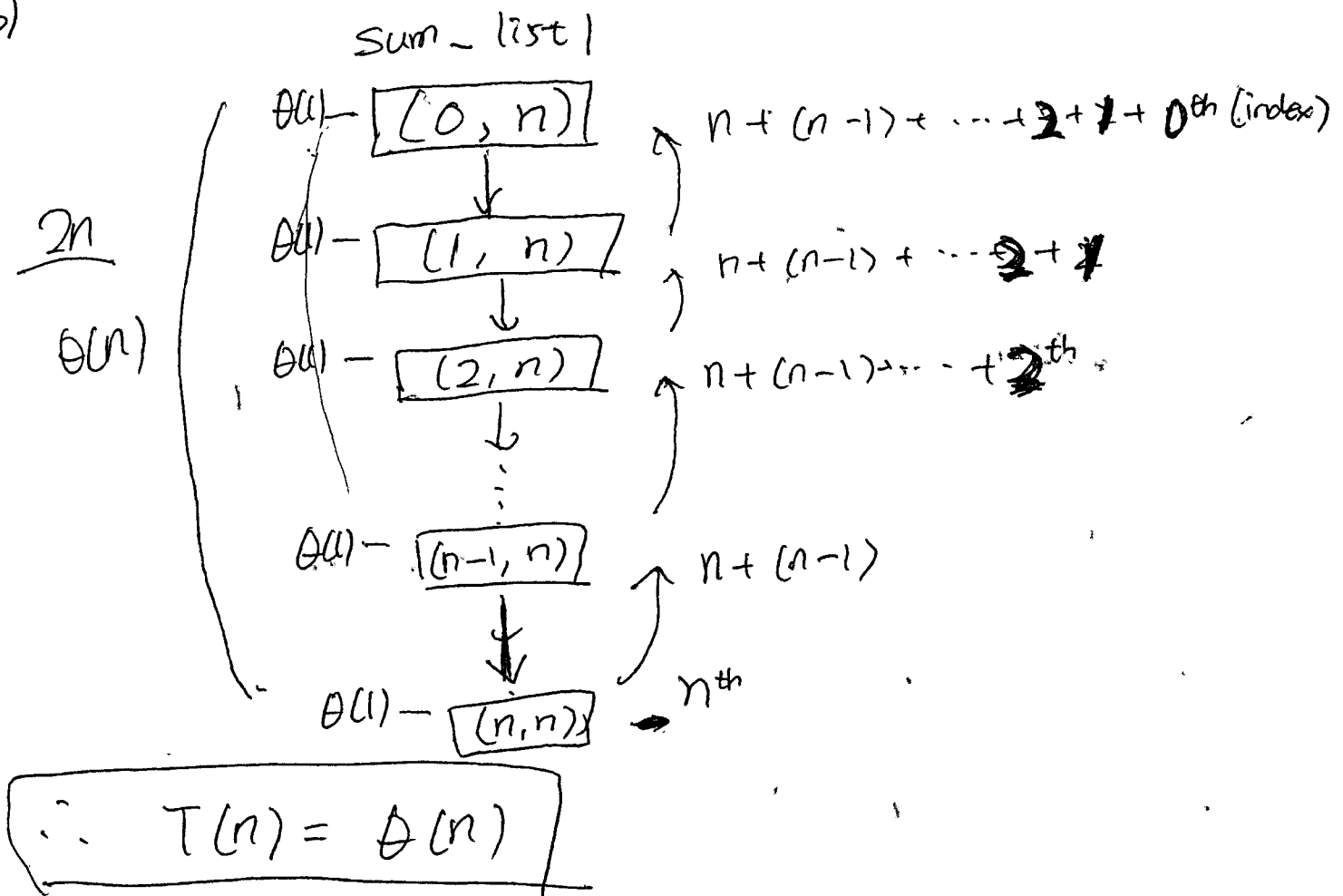
sum_list2, low = 6, high = 6

sum_list2, low = 7, high = 7

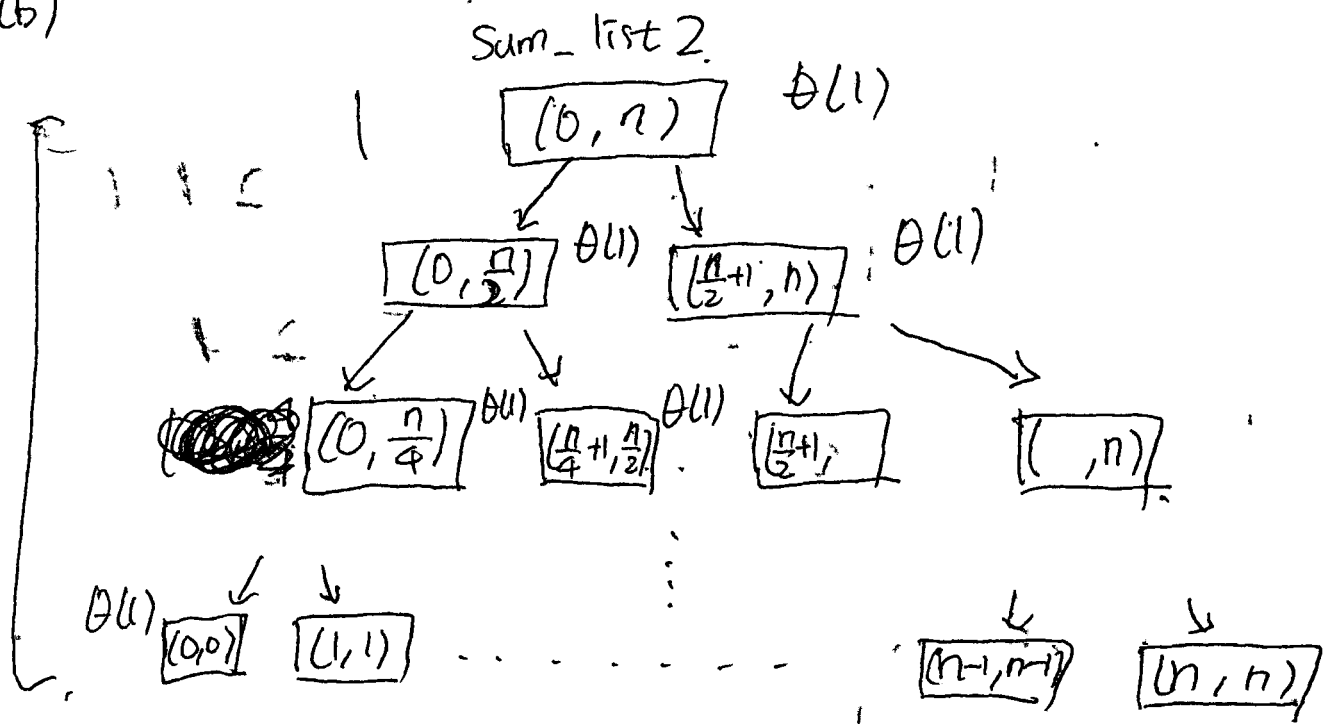
36.

4. recursion tree and total asymptotic running time.

(b)



4. (b)



$$1 + 2 + 4 + 8 + \dots + \frac{n}{2} + n = 2n - 1$$

$$\therefore T(n) = \Theta(n)$$