HW4

1. From Ryan Duffy.

1) Illustrate the operation of PARTITION on the array A = {13, 19, 9, 5, 12, 8, 7, 4, 21, 2, 6, 11}.

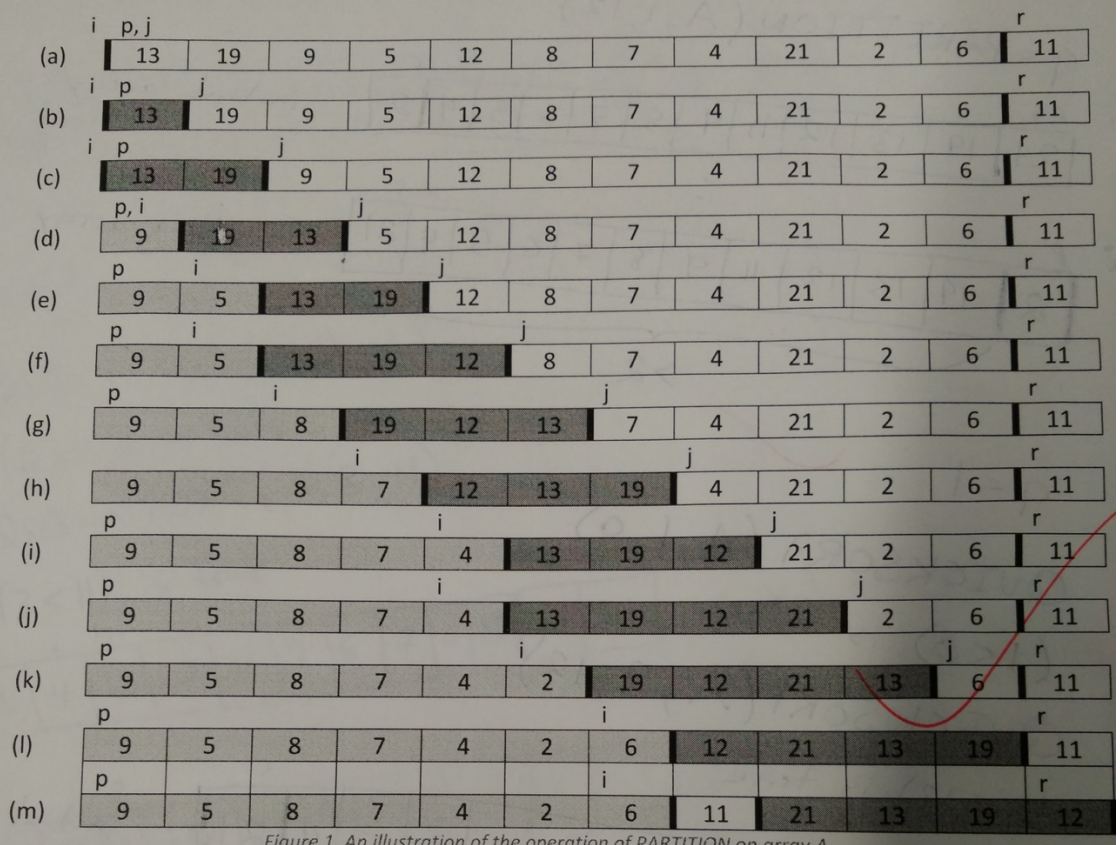


Figure 1. An illustration of the operation of PARTITION on array A.

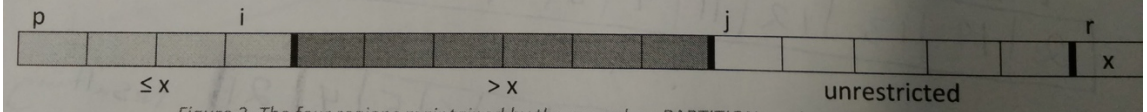
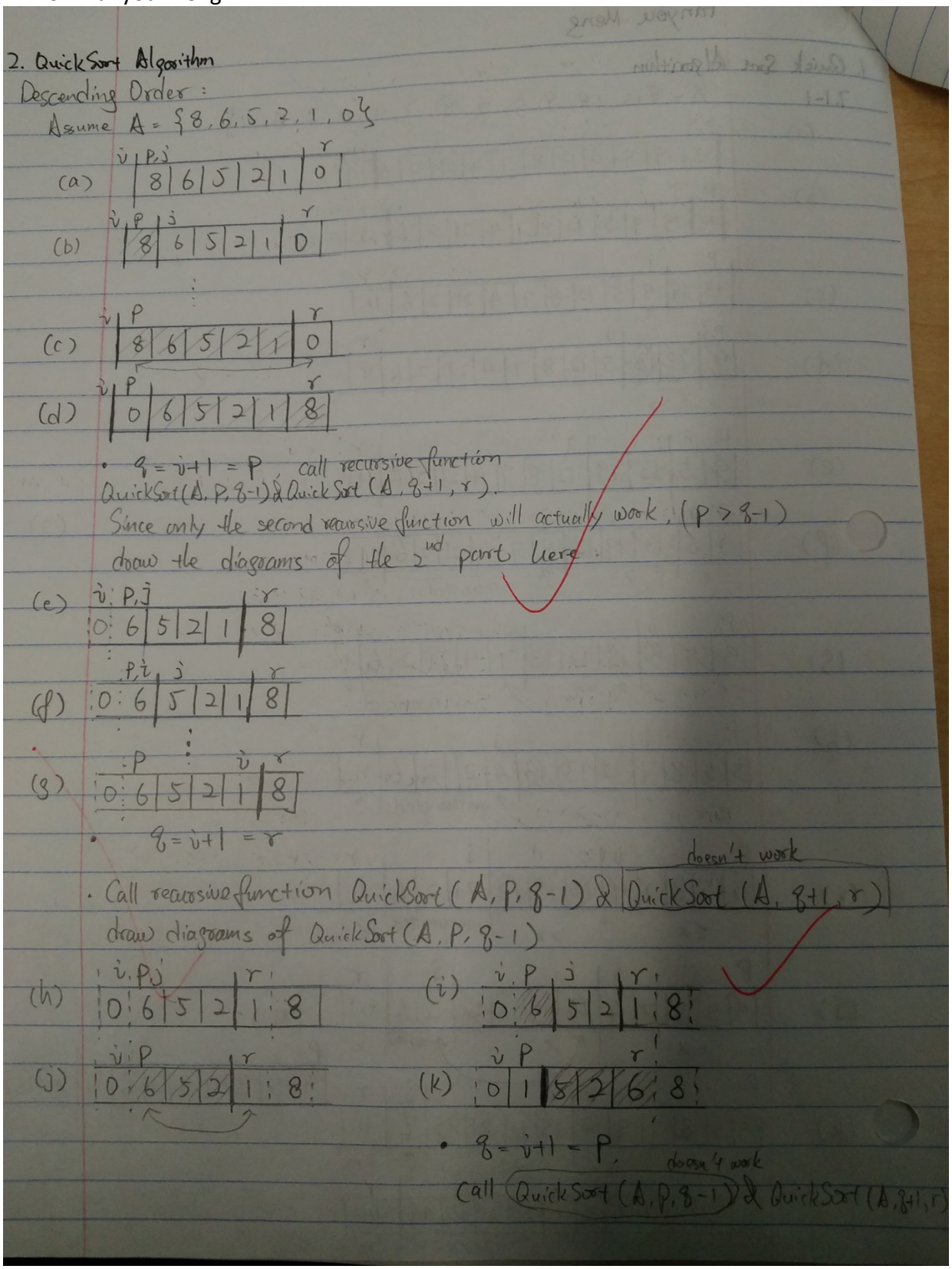


Figure 2. The four regions maintained by the procedure PARTITION on the subarray A[p...r].

```
QUICKSORT(A, p, r)
1.  if p < r
2.          q = PARTITION(A, p, r)
3.          QUICKSORT(A, p, q – 1)
4.          QUICKSORT(A, q + 1, r)
```

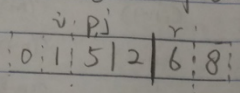
```
PARTITION(A, p, r)
1.  x = A[r]
2.  i = p - 1
3.  for j = p to r -1
4.          if A[j] ≤ x
5.                  i = i + 1
6.                  exchange A[i] with A[j]
7.  exchange A[i +1] with A[r]
8.  return i + 1
```

2. From Fanyou Meng.

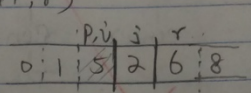


draw diagrams of QuickSort(A, q+1, r)

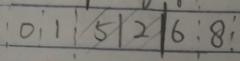
(L)



(M)



(N)

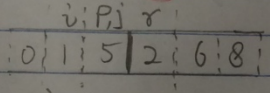


- $q = i+1 = r$
- Call QuickSort(A, P, q-1) & Quick(A, q+1, r)

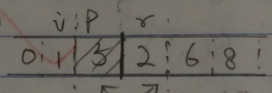
doesn't work

draw diagrams of QuickSort(A, P, q-1)

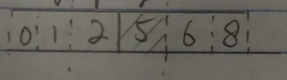
(O)



(P)



(Q)



- $q = i+1 = P$
- Call QuickSort(A, P, q-1) & Quick(A, q+1, r)

doesn't work

- Reach the bottom of recursion.

Array A is |0|1|2|5|6|8| now.

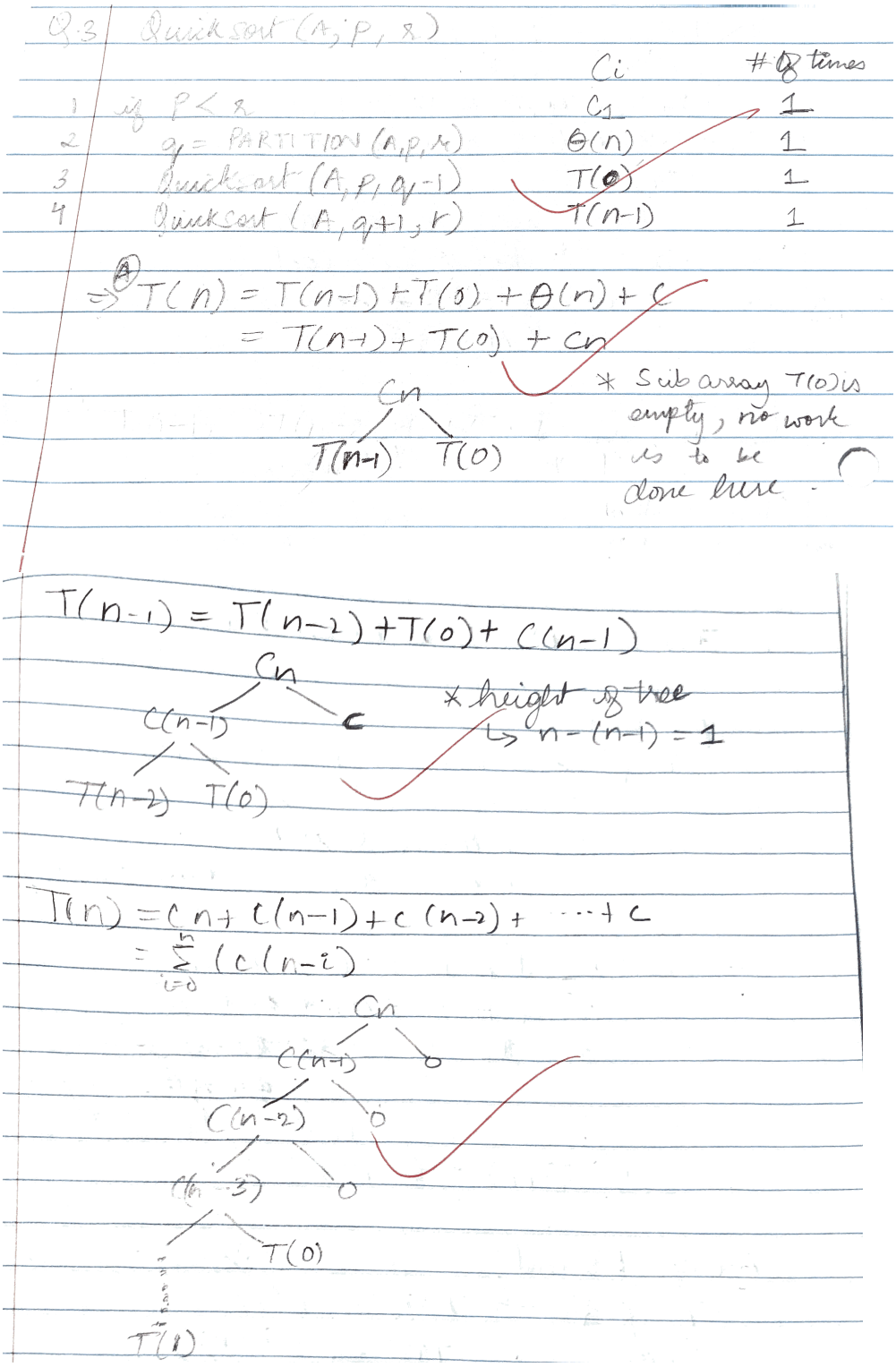
Therefore, if the array is in descending order, q is either p or r.

When the pivot is the current smallest element in the subarray, q is equal to p ; when the pivot is the largest element in the current subarray, q is equal to r.

QuickSort Algorithm Running time

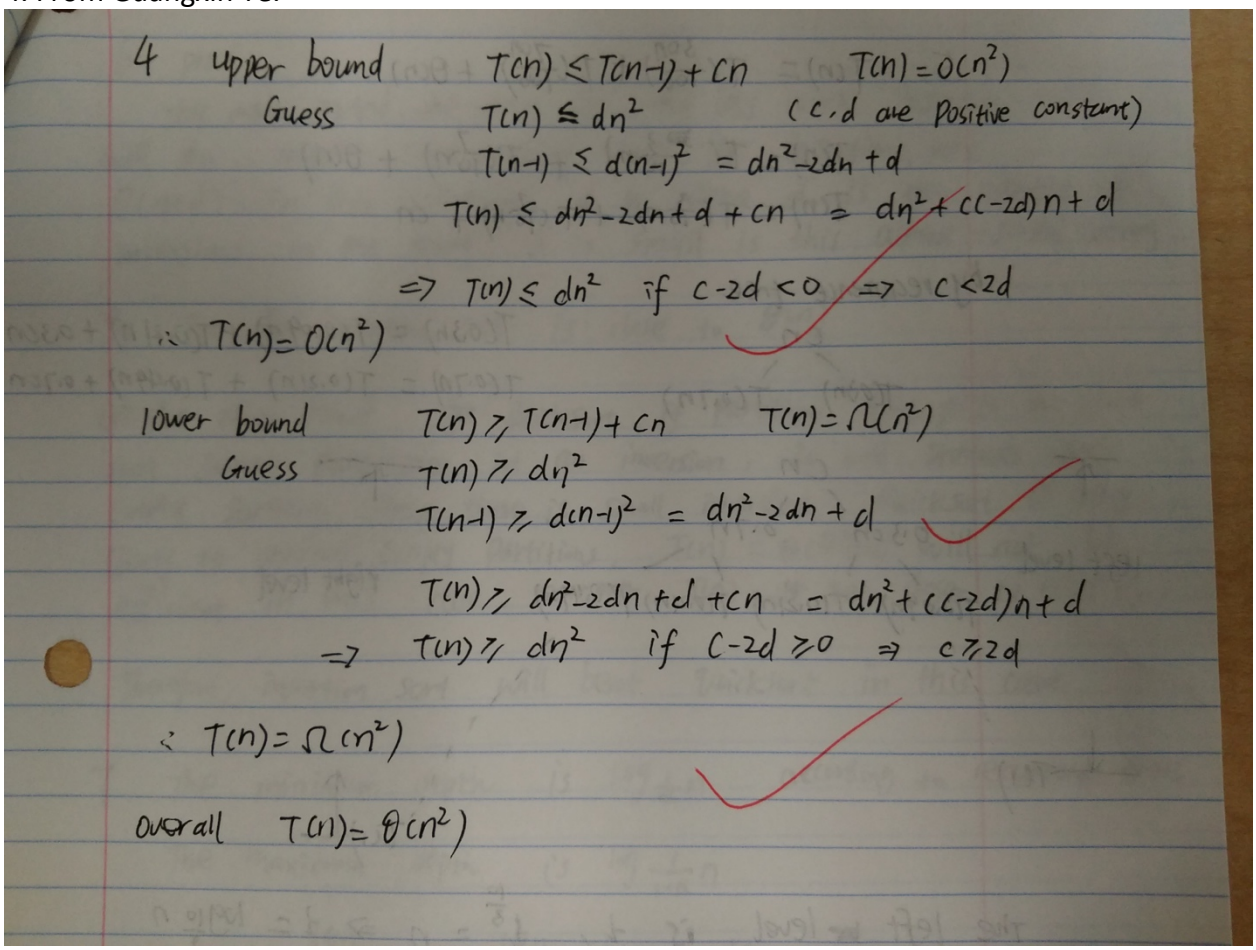
The PARTITION

3. From Chahat Gupta

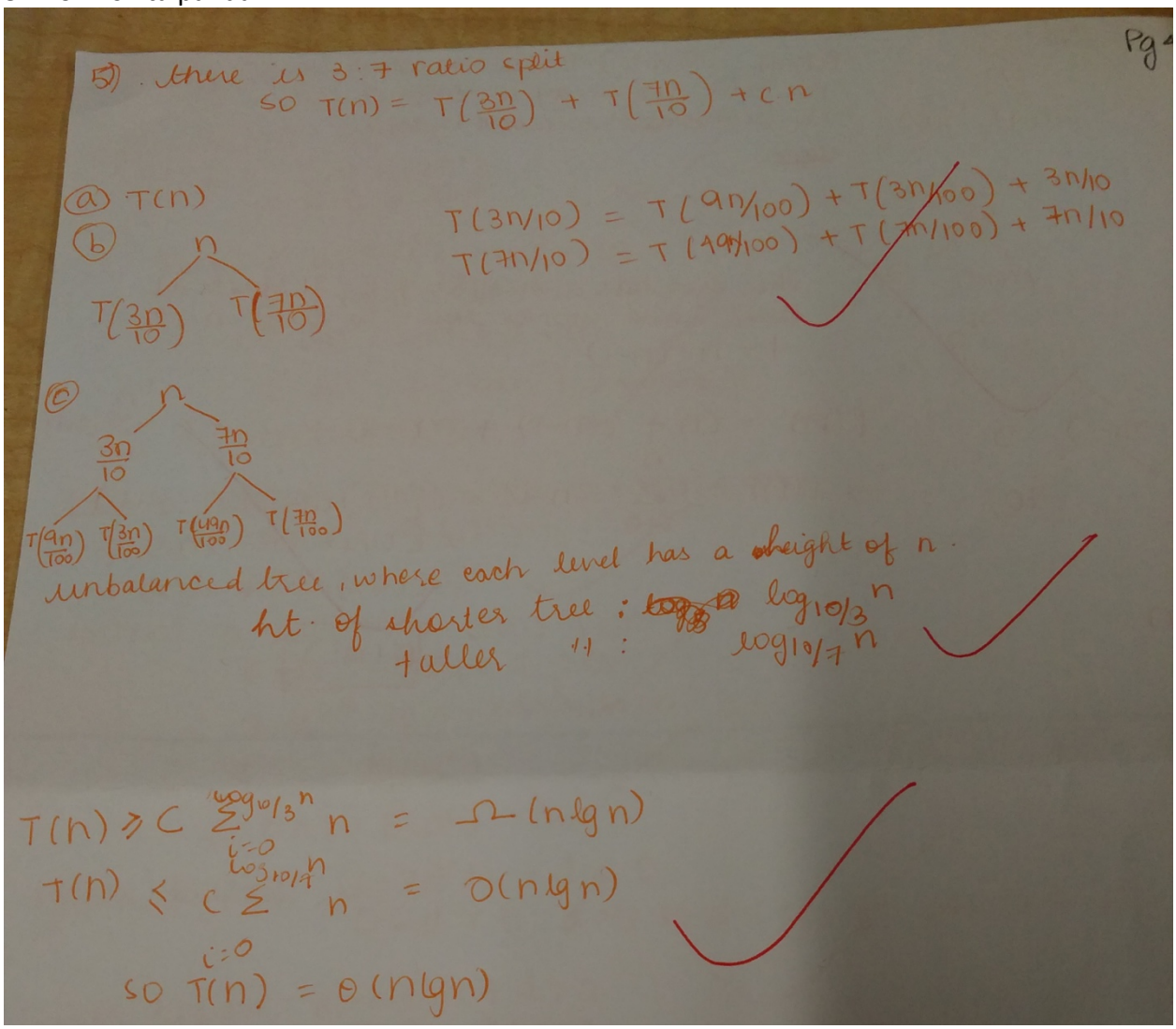


$T(n) = c(n+(n-1)+(n-2)+ \dots + 1) = cn(n+1)/2 = \Theta(n^2)$

4. From Guangxin Ye.

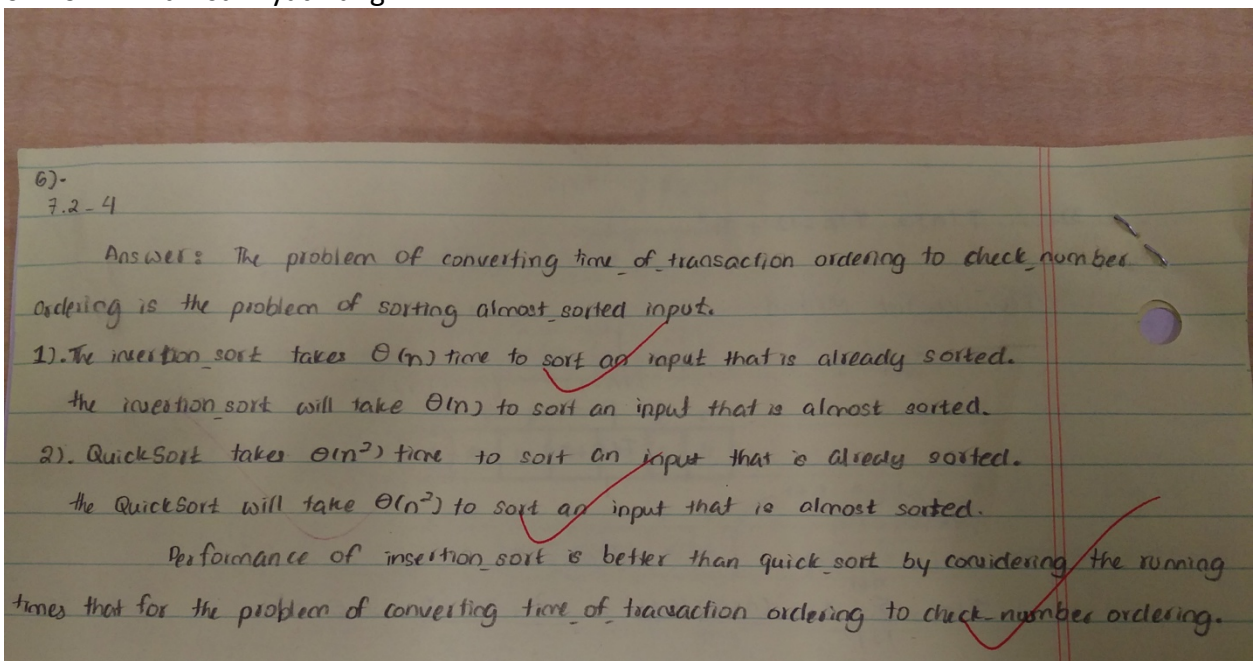


5. From Ishita panda.



May use Theorem: $T(n) = T(\alpha n) + T((1-\alpha)n) + \Theta(n)$, where $0<\alpha \leq 1/2$
$T(n) = \Theta(n \lg n)$

6. From Third Hsu Myat Aung



6)-
7.2-4

Answer: The problem of converting time_of_transaction ordering to check_number ordering is the problem of sorting almost_sorted input.

1). The insertion_sort takes $\Theta(n)$ time to sort an input that is already sorted.
the insertion_sort will take $\Theta(n)$ to sort an input that is almost sorted.

2). QuickSort takes $\Theta(n^2)$ time to sort an input that is already sorted.
the QuickSort will take $\Theta(n^2)$ to sort an input that is almost sorted.

Performance of insertion_sort is better than quick_sort by considering the running times that for the problem of converting time_of_transaction ordering to check_number ordering.

7. From Menaka Gnanavel.