Design and Analysis of Algorithms
CS575, Spring 2023

Theory Assignment 2.3

Due on 3/20/23 (Monday)

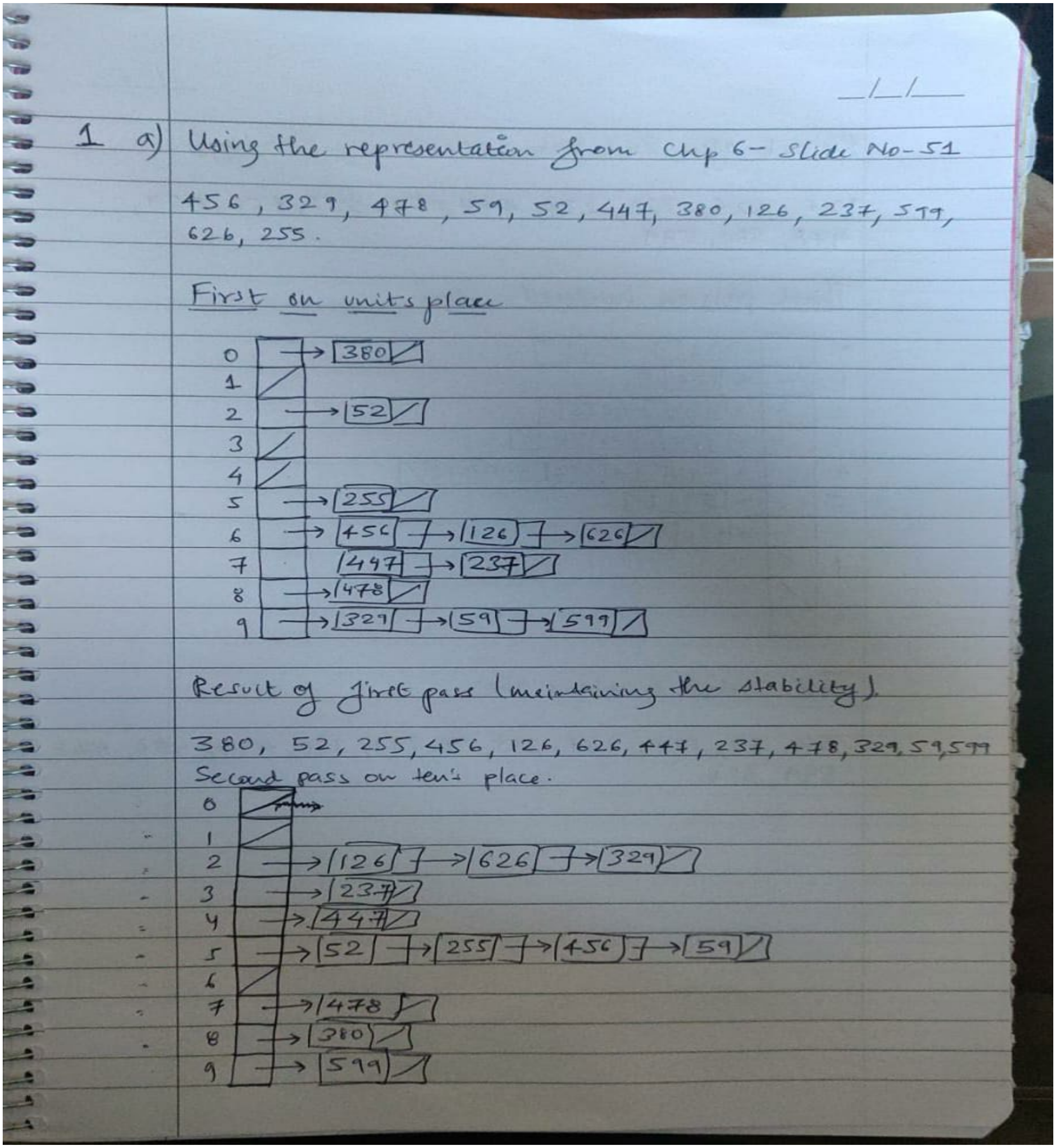
1. (14 points) Use the radix sort algorithm to sort the following numbers. Treat every data as a 3-digit integer.
456, 329, 478, 59, 52, 447, 380, 126, 237, 599, 626, 255.

   a) Draw a figure to show the actions step by step (see example figure in slide 50 or 51 of Ch6-sorting-heap-linear lecture notes) by treating each digit as a "digit". (5 points)

   b) Explain why stable sorting at each step is important. You just need to state that correctness cannot be guaranteed (by giving an example) if you did not apply stable sorting at that step (5 points).

   c) Describe what conditions should be met for radix sort to be O(n)? (4 points)

Solution:
1a.



__/__/____

1 a) Using the representation from Chp 6- Slide No-51

456, 329, 478, 59, 52, 447, 380, 126, 237, 599, 626, 255.

First on units place

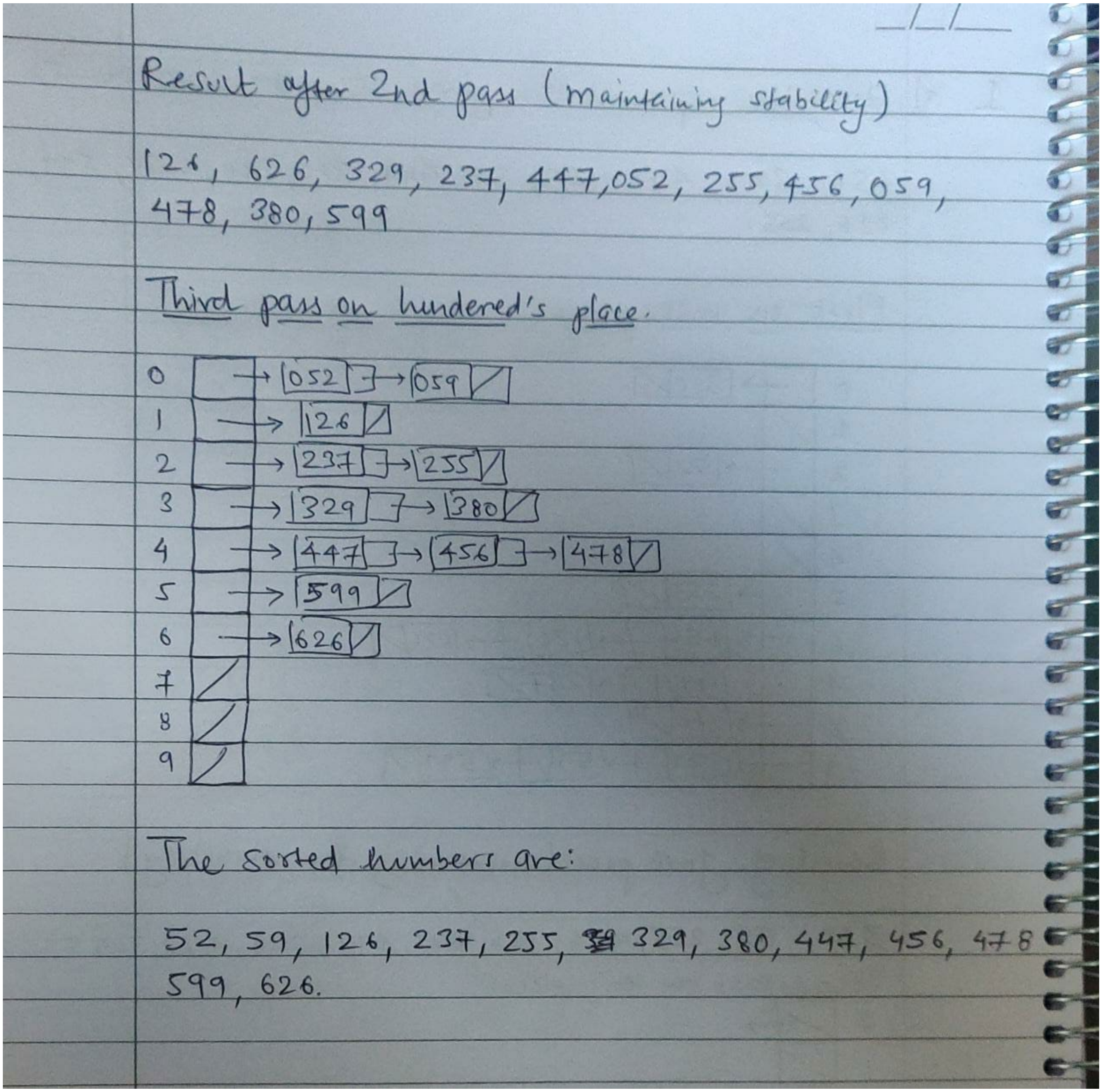
0 → 380
1
2 → 52
3
4
5 → 255
6 → 456 → 126 → 626
7 447 → 237
8 → 478
9 → 329 → 59 → 599

Result of first pass (maintaining the stability).

380, 52, 255, 456, 126, 626, 447, 237, 478, 329, 59, 599

Second pass on ten's place.

0
1
2 → 126 → 626 → 329
3 → 237
4 → 447
5 → 52 → 255 → 456 → 59
6
7 → 478
8 → 380
9 → 599



Result after 2nd pass (maintaining stability)

126, 626, 329, 237, 447, 052, 255, 456, 059, 478, 380, 599

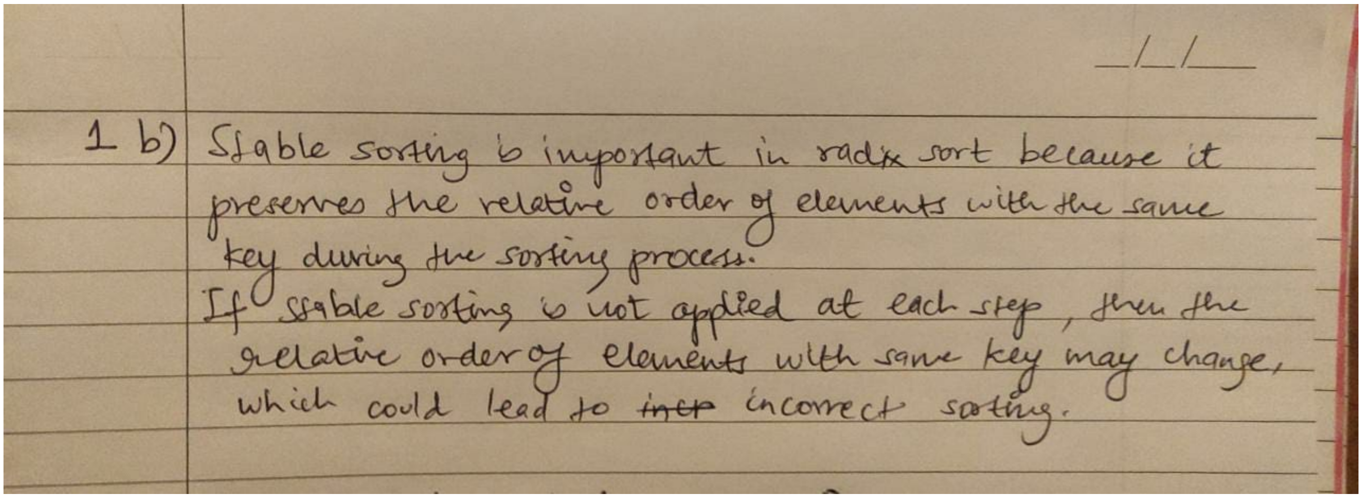
Third pass on hundered's place.

0	052 → 059
1	126
2	237 → 255
3	329 → 380
4	447 → 456 → 478
5	599
6	626
7	
8	
9	

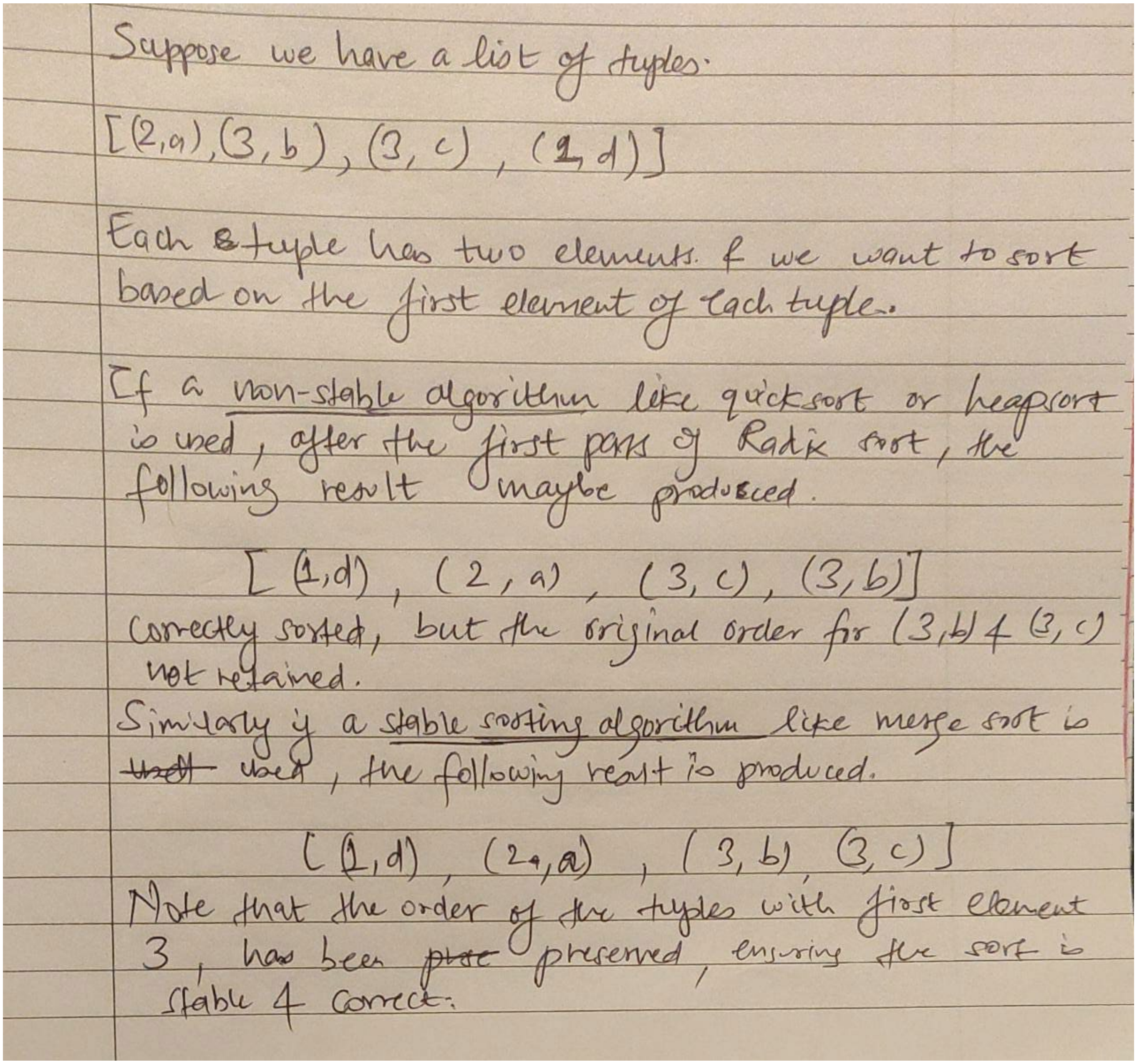
The sorted numbers are:

52, 59, 126, 237, 255, 329, 380, 447, 456, 478 599, 626.

1b.



1 b) Stable sorting is important in radix sort because it preserves the relative order of elements with the same key during the sorting process.
If stable sorting is not applied at each step, then the relative order of elements with same key may change, which could lead to ~~incr~~ incorrect sorting.



Suppose we have a list of tuples.

[(2,a),(3,b),(3,c),(1,d)]

Each tuple has two elements. f we want to sort based on the first element of each tuple.

If a non-stable algorithm like quicksort or heapsort is used, after the first pass of Radix sort, the following result maybe produced.

[(1,d), (2,a), (3,c), (3,b)]

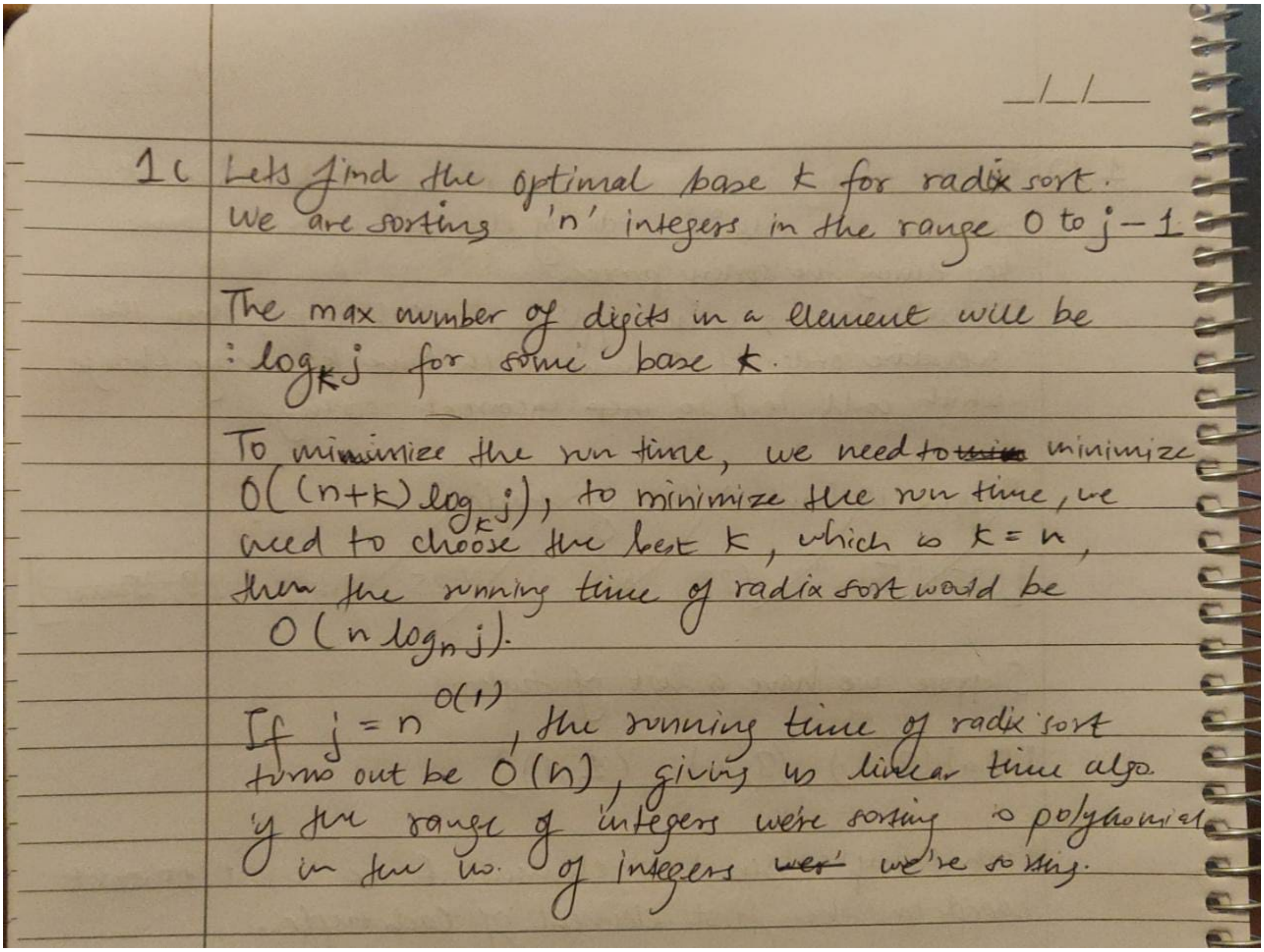
correctly sorted, but the original order for (3,b) & (3,c) not retained.

Similarly if a stable sorting algorithm like merge sort is ~~used~~ used, the following result is produced.

[(1,d), (2,a), (3,b), (3,c)]

Note that the order of the tuples with first element 3, has been ~~plac~~ preserved, ensuring the sort is stable & correct.

1c



_/_/__

1c Lets find the optimal base k for radix sort.
We are sorting 'n' integers in the range 0 to j−1

The max number of digits in a element will be
: $\log_k j$ for some base k.

To minimize the run time, we need to ~~[illegible]~~ minimize
$O((n+k)\log_k j)$, to minimize the run time, we
need to choose the best k, which is $k = n$,
then the running time of radix sort would be
$O(n \log_n j)$.

If $j = n^{O(1)}$, the running time of radix sort
turns out be $O(n)$, giving us linear time algo.
if the range of integers were sorting is polynomial
in the no. of integers ~~wer~~ we're sorting.

1. (14 points) Suppose we want to apply Radix Sort to sort 100,000 4-letter words with each letter taken from the English alphabet (26 letters, all lower cases). Assuming that the running time for sorting *n* elements within range 1..*k* using Counting Sort is 2*n*+2*k*, calculate the running time for each of the following strategies a), b) and c). Show the justification.

   a) treat letters at each of the four positions as a digit. (4 points)

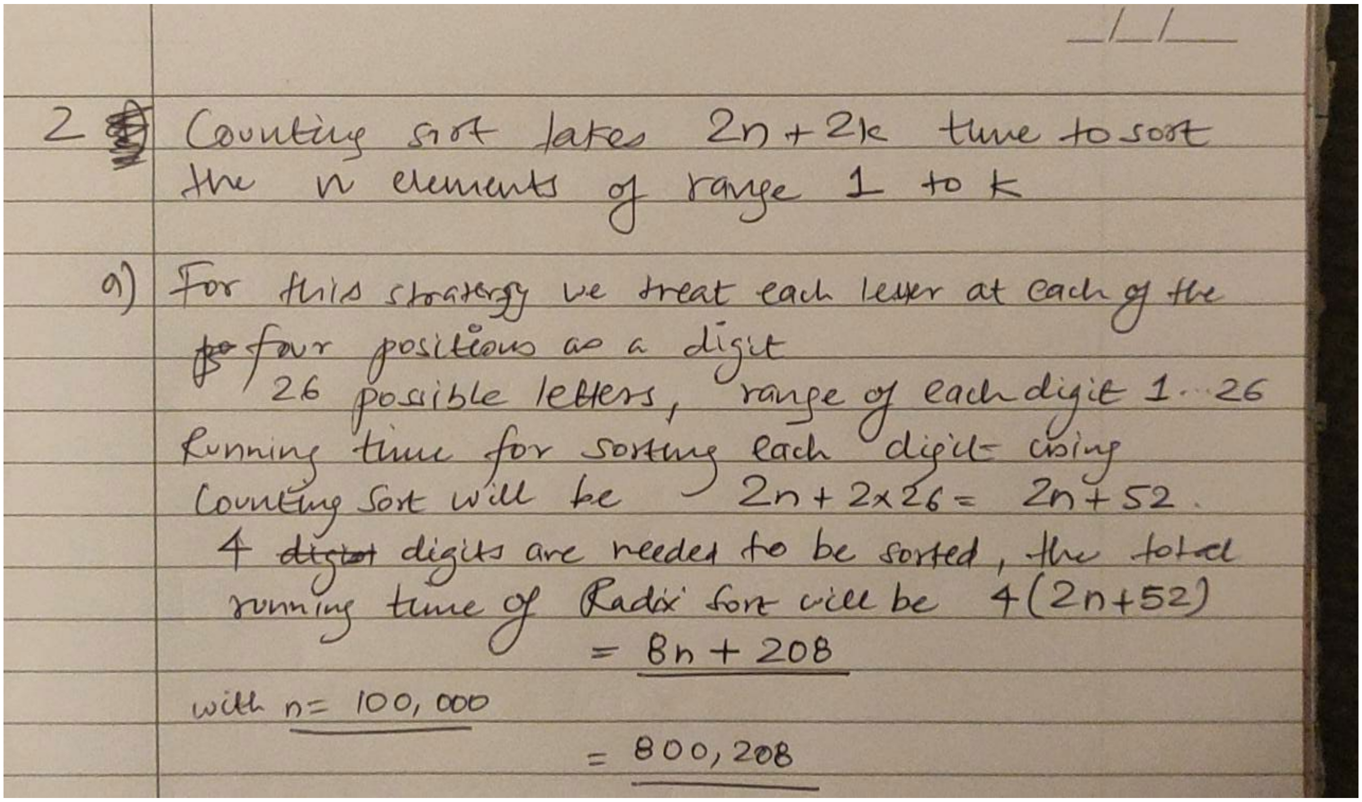
   b) treat 2-letter sub-words at positions 1-2 as a digit and 2-letter sub-words at positions 3-4 as another digit. (4 points)

   c) treat all 4 letters as a digit. (4 points)

   d) which strategy is the best strategy to minimize the running time? (2 points)

Solution:

a.



2 Counting sort takes $2n + 2k$ time to sort the n elements of range 1 to k

a) For this stratergy we treat each letter at each of the four positions as a digit

26 possible letters, range of each digit 1..26

Running time for sorting each digit using Counting Sort will be $2n + 2 \times 26 = 2n + 52$.

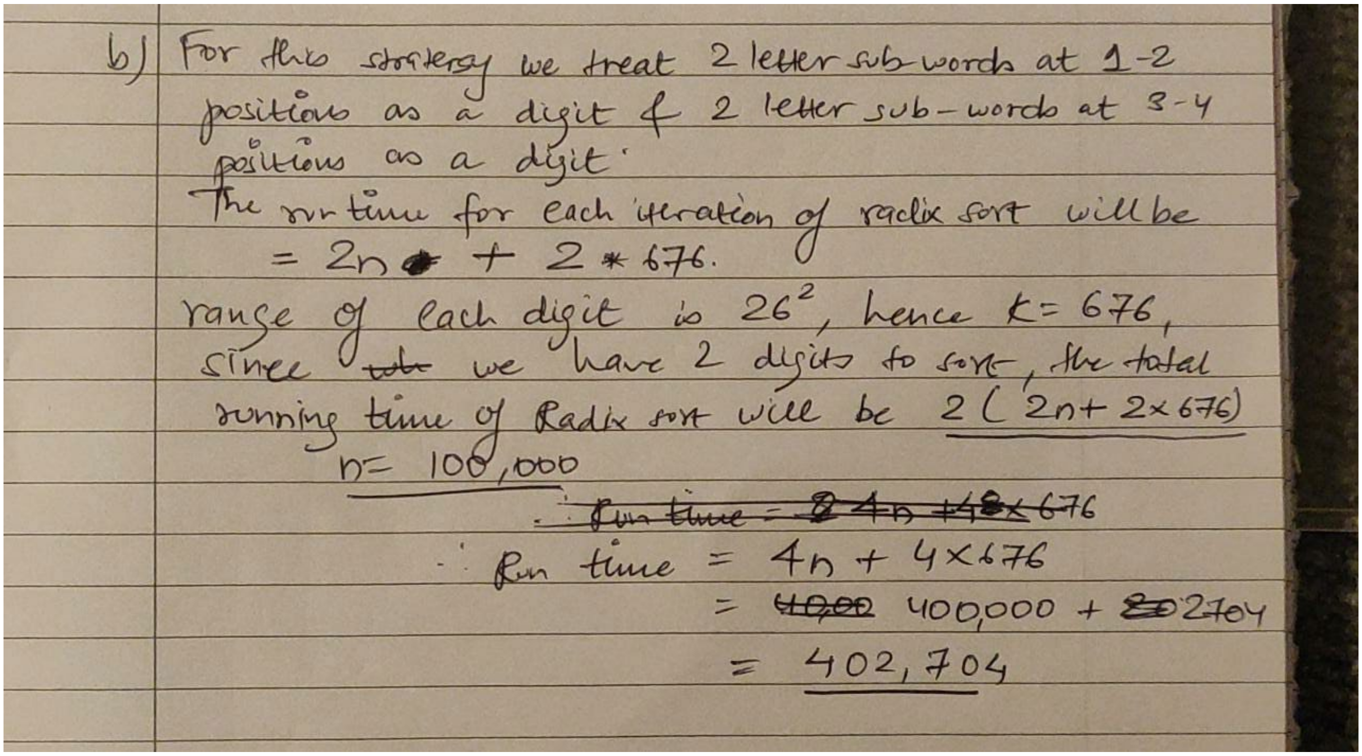
4 ~~digit~~ digits are needed to be sorted, the total running time of Radix sort will be $4(2n+52)$

$= \underline{8n + 208}$

with $\underline{n = 100,000}$

$= \underline{800,208}$

b.



b) For this strategy we treat 2 letter sub-words at 1-2 positions as a digit & 2 letter sub-words at 3-4 positions as a digit.

The run time for each iteration of radix sort will be

$= 2n + 2 * 676.$

range of each digit is 26^2, hence $k = 676$, since ~~tot~~ we have 2 digits to sort, the total running time of Radix sort will be $\underline{2(2n + 2 \times 676)}$

$\underline{n = 100,000}$

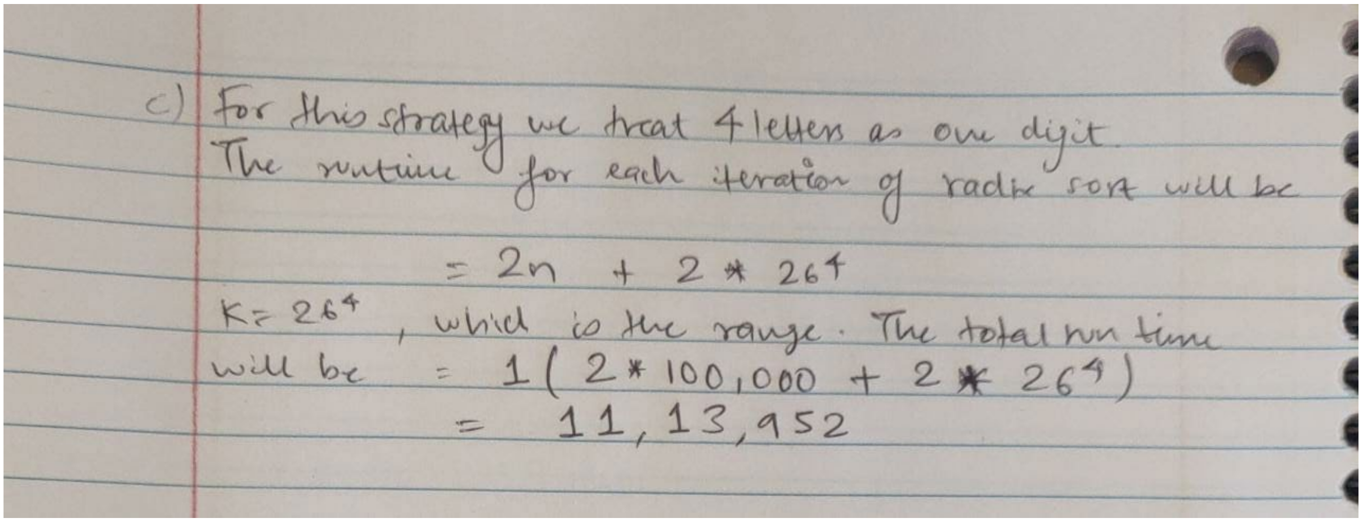
~~∴ Run time = 8 4n + 48x 676~~

∴ Run time $= 4n + 4 \times 676$

$=$ ~~40,00~~ $400,000 +$ ~~80~~ 2704

$= \underline{402,704}$

c.



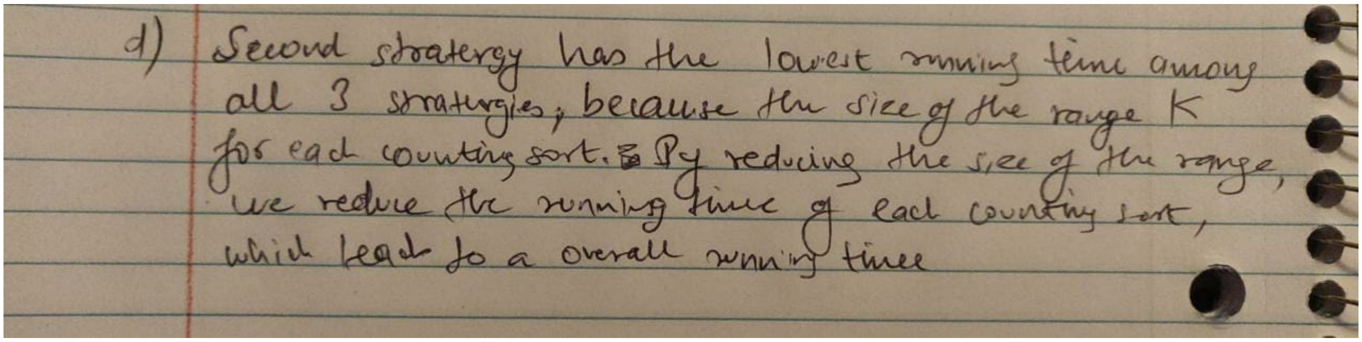
c) For this strategy we treat 4 letters as one digit. The runtime for each iteration of radix sort will be

$$= 2n + 2 * 26^4$$

$K = 26^4$, which is the range. The total run time will be $= 1(2 * 100{,}000 + 2 * 26^4)$

$$= 11{,}13{,}952$$

d.



d) Second straterg has the lowest running time among all 3 straturgies, because the size of the range K for each counting sort. By reducing the size of the range, we reduce the running time of each counting sort, which lead to a overall running timee