Question 1:

$T(n) = 3T(\frac{n}{4}) + 4n$

Substitute once $(n_1 = \frac{n}{4})$

$T(n) = 3[3T(\frac{n}{16}) + 4(\frac{n}{4})] + 4n = 3^2 T(\frac{n}{16}) + 3n + 4n$

Substitute again $(n_2 = \frac{n}{16})$

$T(n) = 3^2[3T(\frac{n}{64}) + 4(\frac{n}{16})] + 7n = 3^3 T(\frac{n}{64}) + 3^2 \frac{4n}{16} + 7n$

$3^2 \cdot \frac{4n}{16} = 9 \cdot \frac{n}{4} = \frac{9n}{4}$

$\hookrightarrow T(n) = 3^3 T(\frac{n}{64}) + \frac{9n}{4} + 7n$

After k substitutions

$$T(n) = 3^k T\left(\frac{n}{4^k}\right) + 4n \sum_{i=0}^{k-1} \left(\frac{3}{4}\right)^i$$

$4^k = n \Rightarrow k = \log_4 n$

$$\hookrightarrow T(n) = 3^{\log_4 n} T(1) + 4n \sum_{i=0}^{\log_4 n - 1} \left(\frac{3}{4}\right)^i$$

$$\sum_{i=0}^{k-1} \left(\frac{3}{4}\right)^i = 4\left[1 - \left(\frac{3}{4}\right)^{\log_4 n}\right]$$

$$\hookrightarrow T(n) = n^{\log_4 3} T(1) + 16n - 16n^{\log_4 3}$$

$$= 16n + (T(1) - 16)n^{\log_4 3}$$

$\log_4 3 = 0.79 < 1$

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

master Theorem

$T(n) = aT(\frac{n}{b}) + f(n)$

$a = 3$

$b = 4$

$f(n) = 4n$

$n^{\log_4 3} = O(n)$

$f(n) = 4n = \Theta(n)$

$n^{\log_4 3} = \Theta(n^{0.792})$

$\left.\begin{matrix} f(n) = 4n = \Theta(n) \\ n^{\log_4 3} = \Theta(n^{0.792}) \end{matrix}\right\} f(n) = \Omega(n^{\log_4 3})$

$af(\frac{n}{b}) \leq cf(n)$

$af(\frac{n}{b}) = 3 \cdot 4(\frac{n}{4}) = 3n$

$3n \leq c(4n)$

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

$$\boxed{T(n) = 16n + (T(1) - 16)n^{\log_4 3}, \text{ so } T(n) = \Theta(n)}$$

Question 2

2.)

Case 1: If $f(n) = O(n^c)$ with $c < \log_b a$,
then $T(n) = \Theta(n^{\log_b a})$

Case 2: If $f(n) = \Theta(n^c)$ with $c = \log_b a$,
then $T(n) = \Theta(n^c \log n)$

Case 3: If $f(n) = \Omega(n^c)$ with $c > \log_b a$
then $T(n) = \Theta(f(n))$

(a) $T(n) = 3T(\frac{n}{5}) + n^2$

$a = 3$
$b = 5$
$f(n) = n^2$

$\log_5 3 \approx 0.68$

$f(n) = n^2$ and $2 > 0.68$
↳ case 3 applies
$\boxed{T(n) = \Theta(n^2)}$

(b) $T(n) = 4T(\frac{n}{3}) + 7n$

$a = 4$
$b = 3$
$f(n) = 7n$

$\log_3 4 \approx 1.26$
$f(n) = O(n^1)$ and $1 < 1.26$
↳ case 1 applies
$\boxed{T(n) = \Theta(n^{1.26})}$

(c) $T(n) = 5T(\frac{n}{4}) + 10$

$a = 5$
$b = 4$
$f(n) = O(1)$

$\log_4 5 \approx 1.16$
$f(n) = O(1)$ and $0 < 1.16$
↳ case 1 applies
$\boxed{T(n) = \Theta(n^{1.16})}$

(d) $T(n) = 9T(\frac{n}{3}) + n^4$

$a = 9$
$b = 3$
$f(n) = n^4$

$\log_3 9 = 2$
$f(n) = n^4$ and $4 > 2$
↳ case 3 applies
$\boxed{T(n) = \Theta(n^4)}$

(e) $T(n) = 6T(\frac{n}{8}) + n^3$

$a = 6$

$b = 8$

$f(n) = n^3$

$\log_8 6 = 0.9$

$f(n) = n^3$ and $3 > 0.9$

↳ case 3 applies

$\boxed{T(n) = \Theta(n^3)}$

<u>Question 3</u>

Group by last letter in alphabetical order:

- A: VEA
- B: JOB
- D: USD, DOD, CAD *(order preserved)*
- E: VEE
- G: FIG, PIG, DOG
- L: COL, LOL, TSL
- N: SUN
- P: CAP
- R: CAR
- S: VIS
- T: RAT
- W: ROW, WOW, LOW
- X: COX, LOX
- Y: JPY

<u>After Pass 1, the list becomes:</u>

VEA, JOB, USD, DOD, CAD, VEE, FIG, PIG, DOG, COL, LOL, TSL, SUN, CAP, CAR, VIS, RAT, ROW, WOW, LOW, COX, LOX, JPY

Sort by the Middle Character

- A: CAD, CAP, CAR, RAT
  (Order from Pass 1: CAD (5th), CAP (14th), CAR (15th), RAT (17th))

- E: VEA, VEE
- I: FIG, PIG, VIS
- O: JOB, DOD, DOG, COL, LOL, ROW, WOW, LOW, COX, LOX
- P: JPY
- S: USD, TSL
- U: SUN

After Pass 2, the list becomes:

CAD, CAP, CAR, RAT, VEA, VEE, FIG, PIG, VIS, JOB, DOD, DOG, COL, LOL, ROW, WOW, LOW, COX, LOX, JPY, USD, TSL, SUN

Pass 3: Sort by the First Character

- C: CAD, CAP, CAR, COL, COX
- D: DOD, DOG
- F: FIG
- J: JOB, JPY
- L: LOL, LOW, LOX
- P: PIG
- R: RAT, ROW
- S: SUN
- T: TSL
- U: USD
- V: VEA, VEE, VIS
- W: WOW

After Pass 3, the list becomes:
CAD, CAP, CAR, COL, COX, DOD, DOG, FIG, JOB, JPY, LOL, LOW, LOX, PIG, RAT, ROW, SUN, TSL, USD, VEA, VEE, VIS, WOW

Question 4:

4.)

Hash Table

Initial size (M): 13

Resized (M): 29

primary hash function:

$$h1(key) = \left(\frac{(key+19)(key+11)}{15} + key\right) \bmod M$$

Key	Home Slot	Collisions	Probe Sequence	Final Slot
25	0	0		0
14	4	0		4
9	7	0		7
7	12	0		12
5	4	1	(4+5) mod 13 = 9	9
3	10	0		19
0	0	infinite loop	resize + rehash	
21	19	0		19
6	5	0		5
33	11	1	(11+33) mod 29 = 15	15
25	14	1	(14+52) mod 29 = 8	8
42	25	0		25
24	8	1	(8+42) mod 29 = 21	21
107	25	2	(25+701) mod 29 = 1, (1+701) mod 29 = 6	6

Final Hash Table

Slot	key
0	—
1	5
2	—
3	—
4	—
5	6
6	107
7	—
8	25
9	7
10	—
11	14
12	—
13	0
14	25
15	33
16	—
17	9
18	—
19	21
20	—
21	24
22	—
23	3
24	—
25	42
26	—
27	—
28	—

<u>Question 7:</u>

RadixSorting Algorithm Analysis

- Time Complexity: O(M(N + K))
    - N = number of elements in an array
    - M = maximum length of an element/maximum number of digits or characters
    - K = range of characters
    - Radix sort processes M passes, and each pass uses counting sort which takes O(N + K) time.

- Space Complexity: O(N + K)
    - Where N is the number of elements in the input array and K is the range of digits/characters. It requires extra storage for the output array and counting array.

WordPattern Algorithm Analysis

- Time Complexity: O(n)
    - The function splits the string s into an array, which takes O(n) where n is the length of s.
- Space Complexity: O(n)
    - The split words array takes up O(n) space, where n is the length of the pattern. Two hashmaps store character-word pairs, which require up to O(n) (where n is the length of pattern) space each in the worst case.