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

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

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

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

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

$= 3^i T(\frac{n}{4^i}) + n \cdot \sum_{k=1}^{i} (3^{k-1} \cdot 4^{2-k})$

$T(n) = 3^{\log_4 n} T(1) + n \cdot \sum_{k=1}^{\log_4 n} (3^{k-1} \cdot 4^{2}$

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

$a = 3 \quad b = 4 \quad d = 1$

$b^d = 4 > 3$

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

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

$a = 3 \quad b = 5 \quad d = 2$

$b^d = 5^2 > 3$

$T(n) = \Theta(n^2)$

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

$a = 4 \quad b = 3 \quad d = 1$

$b^d = 3^1 < 4$

$T(n) = \Theta(n^{\log_3 4})$

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

$a = 5 \quad b = 4 \quad d = 0$

$b^d = 4^0 < 5$

$T(n) = \Theta(n^{\log_4 a})$

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

$a = 9 \quad b = 3 \quad d = 4$

$b^d = 3^4 > 9$

$T(n) = \Theta(n^4)$

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

$a = 6 \quad b = 8 \quad d = 3$

$b^d = 8^3 > 6$

$T(n) = \Theta(n^3)$

3. Radix Sort

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

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

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

Sorted by third letter (fully sorted)
CAD, CAP, CAR, COL, COX, DOD, DOG, FIG, JOB, JPY, LOL, LOW, LOX, PIG, RAT, ROW, SUN TSL, USD, VEA, VEE, VIS WOW

4. Double Hashing

key: 25 initial: 0 collisions: 0
key: 14 initial: 4 collisions: 0
key: 9 initial: 7 collisions: 0
key: 7 initial: 12 collisions: 0
key: 5 initial: 4 collisions: 1 (4, 9)
key: 3 initial: 10 collisions: 0
key: 0 initial: 0 collisions: 0

this causes an infinite loop, so we re-size and rehash
M = 29

key: 25 initial: 14 collisions: 0
key: 14 initial: 11 collisions: 0
key: 9 initial: 17 collisions: 0
key: 7 initial: 9 collisions: 0
key: 5 initial: 1 collisions: 0
key: 3 initial: 23 collisions: 0

key: 0 initial: 13 collisions: 0
key: 21 initial: 19 collisions: 0
key: 6 initial: 5 collisions: 0
key: 33 initial: 11 collisions: 1 (11, 15)
key: 25 initial: 14 collisions: 1 (14, 8)
key: 42 initial: 25 collisions: 0
key: 24 initial: 8 collisions: 1 (8, 21)
key: 107 initial: 25 collisions: 2 (25, 1, 6)

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

7. Algorithm Analysis

Problem 4: The worst case requires re-sizing and rehashing a full hash table, so the algorithm is O(n) where n is the number of slots in the table.

Problem 5: The lexicographic radix sort is still O(n) because the logic is fundamenally the same.

Problem 6: Time is O(n) where n = size of the input string s. The for loop visits each character in the length of s once. Space is O(m) where m = size of the input string pattern. If every character in the pattern is unique, the hash map made will have m key-value pairs. No variables are declared in the for loop.