Assignment 5

Kevin Gonzalez date / /

Repeated Substitution

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

Second expand: $T(n) = 3[3T(\frac{n}{16}) + 4(\frac{n}{4})] + 4n$

$= 9T(\frac{n}{16}) + 12(\frac{n}{4}) + 4n$

Expand Scratch

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

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

Third Expand:

$T(n) = 9[3T(\frac{n}{64}) + 4(\frac{n}{16})] + 12(\frac{n}{4}) + 4n$

$T(n) = 27T(\frac{n}{64}) + 36(\frac{n}{16}) + 12(\frac{n}{4}) + 4n$

This can be simplified: $3^k T(\frac{n}{4^k}) + (4n + 3n + \frac{9n}{4} + \frac{27n}{16} + \dots)$

Recurrence term

$S_k = 4n \sum_{i=0}^{k-1} (\frac{3}{4})^i$

Now using geometric formula: $S_k = \sum_{i=0}^{k} (\frac{3}{4})^i = \frac{1 - (3/4)^{k+1}}{1 - 3/4_k}$

$= \frac{1 - (3/4)}{1/4}$

$= 4(1 - (\frac{3}{4})^k) \rightarrow 4n \cdot 4(1 - (\frac{3}{4})^k) = 16n(1 - (\frac{3}{4})^k)$

Base case is $\frac{n}{4^k} = 1 \rightarrow k = \log_4 n$

$16n(1 - (\frac{3}{4})^k) \approx 16n$

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

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

$a = 3$

$b = 4$

$f(n) = 4n$

$\log_4(3) \approx .793$

$4n = n^1 \rightarrow n^1 > n^{.793}$

Since $c > \log_b(a)$, then Case 3 of master theoram:

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

date / /

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

$a=3$ $\quad \log_5(3) \approx \frac{\log_{10}(3)}{\log_{10}(5)} = \frac{.477}{.644} \approx .682$

$b=5$ $\quad n^2 \doteq n^2$

$f(n)=n^2$ $\quad$ Now compare: $3 < 5^2$

since $c > \log_b a$, then:

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

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

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

$a=4$ $\quad \log_3(4) = \frac{\log_{10}(4)}{\log_{10}(3)} = \frac{.602}{.477} \approx 1.261$

$b=3$ $\quad$ Now compare: $4^1 > 3^{1.261}$

$f(n) = 7n = n$ $\quad$ Since $c < \log_b a$, then:

$T(n) = \theta(n^{\log_b a})$

$T(n) = \theta(n^{1.261})$ or $T(n) = \theta(n^{\log_3 4})$

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

$a=5$ $\quad \log_4(5) \approx 1.161$

$b=4$ $\quad$ Now compare: $5 > 4^1$

$f(n) = 10 = O(1)$ $\quad$ Since $c < \log_b a$, then:

$T(n) = \theta(n^{\log_b a})$

$T(n) = \theta(n^{1.161})$ or $T(n) = \theta(n^{\log_4(5)})$

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

$a=9$ $\quad \log_3(9) = 2$

$b=3$ $\quad$ Now compare: $9 < 3^4$

$f(n) = n^4$ $\quad$ Since $c > \log_b a$, then:

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

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

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

$a=6$ $\quad \log_8(6) \approx .862$

$b=8$ $\quad$ Now compare: $6 < 8^3$

$f(n) = n^3$ $\quad$ Since $c > \log_b a$, then:

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

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

date / /

3. To sort - CAP, COL, USD, SUN, JPY, VEE, ROW, JOB, COX, LOL, RAT, WOW, DOD, CAR, FIG, PIG, VIS, LOW, LOX, VEA, CAD, DOG, TSL - using Radix-Sort.
We must sort by the 3rd letter first, then 2nd, finally 1st.
3rd letter | 2nd letter | 1st letter

	3rd letter	2nd letter	1st letter
A	VEA		
B	JOB		
C		CAD, CAP, CAR, RAT	CAB, CAP, CAR, COL, COX
D	USD, DOD, CAD		DOD, DOG
E	VEE	VEE, VEA	
F			FIG
G	FIG, PIG, DOG		
H			
I		FIG, PIG, VIS	
J			JOB, JPY
K			
L	COL, LOL, TSL		LOL, LOW, LOX
M			
N	SUN		
O		COL, ROW, JOB, COX, LOL, WOW, DOD, LOW, LOX, DOG	
P	CAP	JPY	PIG
Q			
R	CAR		RAT, ROW
S	VIS	USD, TSL	SUN
T	RAT		TSL
U		SUN	USD
V			VEE, VEA, VIS
W	ROW, WOW, LOW		WOW
X	COX, LOX		
Y	JPY		
Z			

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

date / /

4. Following the hash function h1(), we conderive this hash computation:

$$h1(Key) = \left(\frac{(Key+19)\cdot(Key+11)}{15} + Key \right) \bmod 13$$

A reverse secondary function (for double hash): h2(Key) = Reverse(key)

Collision fix woulde

Key	
25	$-((25+19)\cdot(25+11))/15+25) \% 13 = 130 \% 13 = 0$
14	$-((14+19)\cdot(14+11))/15+14) \% 13 = 69 \% 13 = 4$
9	$-((9+19)\cdot(9+11))/15+9) \% 13 = 46 \% 13 = 7$
7	$-((7+19)\cdot(7+19))/15+7) \% 13 = 38 \% 13 = 12$
5	$-((5+19)\cdot(5+11))/15+5) \% 13 = 30 \% 13 = 4$
Reverse —	$(4 + \text{Reverse}(5)) \% 13 = 4+5 \% 13 = 9$
	$3-((3+19)\cdot(3+11)/15 \% 13 = 23 \% 13 = 10$
	$0-((0+19)\cdot(0+11)/15 \% 13 = 13 \% 13 = 0$
Reverse —	$(0+1) \% 13 = 1$ (collision, resize)
	$21-((21+19)\cdot(21+11)/15 \% 13 = 106 \% 13 = 2$
	$6-((6+19)\cdot(8+11)/15 \% 13 = 34 \% 13 = 8$
	$33-((33+19)\cdot(33+11)/15 \% 13 = 185 \% 13 = 3$
	$25-((25+19)\cdot(25+11)/15 \% 13 = 130 \% 13 = 0$
Reverse —	$(1+52) \% 13 = 0$
—	$(0+1) \% 13 = 1$
—	$(0+2) \% 13 = 2$
—	$(0+3) \% 13 = 3$
—	$(0+4) \% 13 = 4$
✓	$(0+5) \% 13 = 5$

Probe	Collision	Slot
0	0	0
4	0	4
7	0	7
12	0	12
9	1	9
10	0	10
1	1	1
2	0	2
8	0	8
3	0	3

due to thashing,
Infinite Loop!!!!

Resize table $13 \cdot 2 = 26$

date / /

$42-((42+19)\cdot(42+11)\%15+42)\%26=23$

$24-((24+19)\cdot(24+11)\%15+24)\%26=20$

$107-((107+19)\cdot(107+11)\%15+107)\%26=6$

Probe	Collision	Slot
23	0	23
20	0	20
6	0	6

date / /

7. Hashing: Time complexity: In the worst case it is O(n) where n is the number of element stored in the hash table. Rehashing also takes O(n) since you copy elements to new table, O therwise, it is O(1) on average.
Space Complexity: O(n) where n is the number of keys stored.

Radix Sort: Time Complexity: O(n·m) where n is the number of strings and m is the max length of a string of an array.
Space Complexity: O(n) where n is the number of strings in the input array. We chop K=256 b/c n is biggest growing variable.
O(n+256)

Word Pattern: Time Complexity: O(n) where n is the length of the string
Space Complexity: O(n) where n is the number of words in the string input.