CS590 - Report

HW - 2

Prithiv Dev Devendran

10453922

Part 1(20 Points)

1) Our guess $T(n) = O(n \log n)$

$T(n) \leq C((n-3) \log (n-3)) + 3\log n$ $T(n) \leq c(3(n-3) \log(n/3-1) + 3\log n$

$T(n) \leq c(3(n-3) \log(n/3-1) + 3\log n$ $T(n) \leq c(n-3) \log n + 3\log n$

$T(n) \leq c\, n\log n - 3c \log n + 3\log n$ $T(n) \leq c\, n\log n$

Hence, our assumption holds good for c>3.

2) Our guess $T(n) = O(n \log 3^4)$ $T(n) \leq cn^{\log 3^4} + n$

$T(n) \leq 4c(n/3) \log 3^4 + n$ $T(n) = cn^{\log 3^4} + n$

Our assumption fails here.

Now our guess is $T(n) \leq cn\log 3^4 - dn$,

$T(n) \leq 4(c(n/3) \log 3^4 - dn/3) + n = 4(cn\log 3^4/4 - dn/3) + n$
$= cn\log 3^4 - 4/3dn + n \leq cn\log 3^4 - dn$

Where our assumption holds good for $d \geq 3$

3) Our guess $n = \text{sum } (k=0 \text{ to } \log 2^n)\ c2^k$

$T(n) = 1/ (1-1/2k-1/4k^2-1/8k^3)$ sum ($k=0$ to $\log 2^n$) $c2^k$ By *inversing*

$T(n)=1-1/2k-1/4k^2-1/8k^3$
For lower bound, by substituting the value in equation

$T(n) \leq 2^{\log 2^n} (1-1/2k-$....................................) holds good for the equation.

For upper bound, by substituting the value in the equation
$T(n) \leq 2^{\log 2^n} + 1 (1-1/2k-$......................) holds good for the equation.

This proves our assumption $0(2^{\log 2^n})$ holds good, Therefore $T(n) = O(2^{\log 2^n}) = O(n)$

4) Our assumption is to show $T(n) \le cn^2$ for some c.

we have $T(n) = 4T(n/2) + n$
$\le 4(c(n/2)^2) + n$

$= cn^2 + n$
which doesn't satisfy to be less than cn^2 for any $c > 0$.

Our approach now

$T(n) \le cn^2 - n$. $T(n) = 4T(n/2) + n$

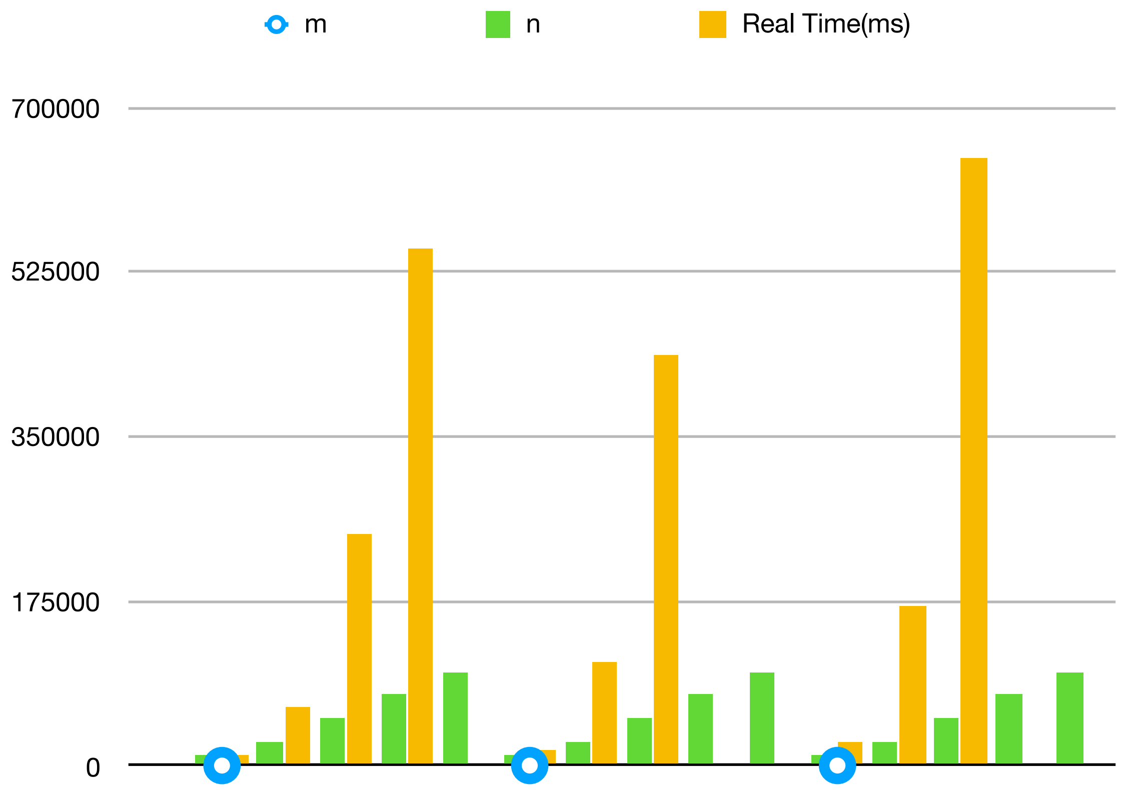
$\le 4(c(n/2)^2 - n) + n = cn^2 - 4cn + n$

$\le cn^2$ for $c \ge 1/4$

Part 2(80 Points)

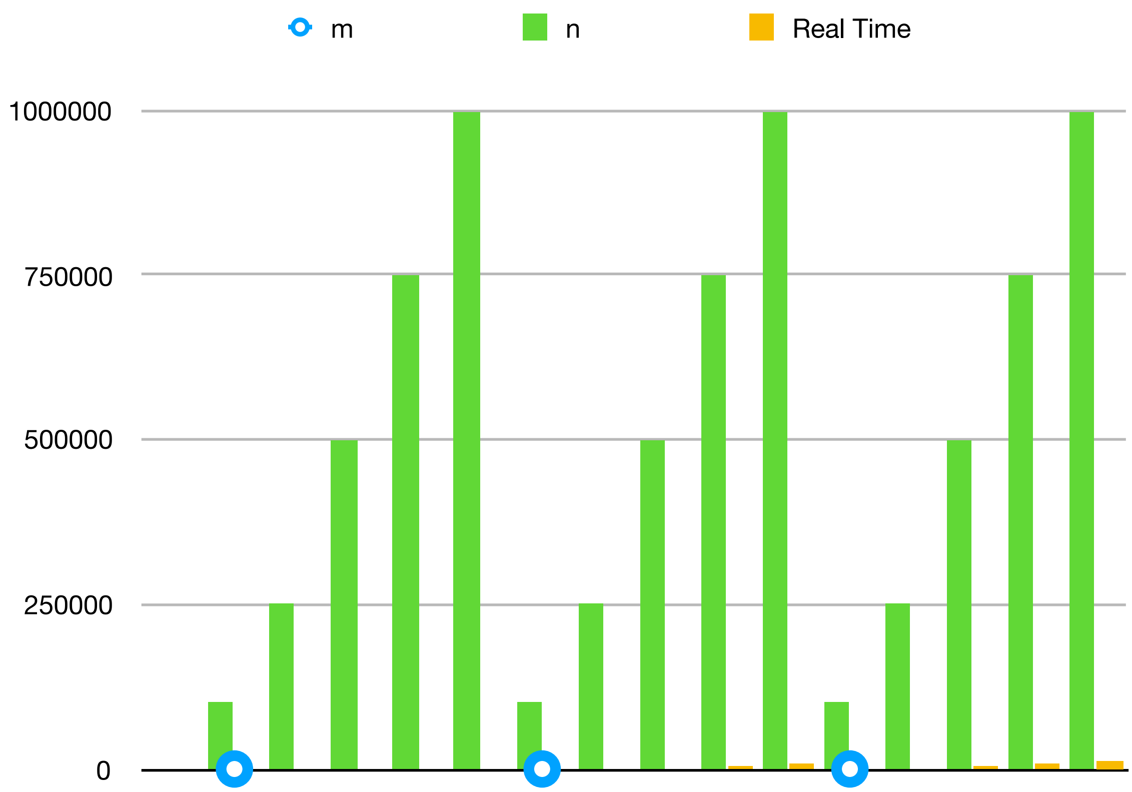
Radix Sort (Insertion Sort)		
m	n	Real Time(ms)
25	10000	9601
	25000	60773
	50000	246064
	75000	548400
	100000	>10 mins
50	10000	17433
	25000	110640
	50000	435490
	75000	>10 mins
	100000	>10 mins
75	10000	25750
	25000	169198
	50000	646551

Radix Sort (Insertion Sort)		
	75000	>10 mins
	100000	>10 mins



Radix Sort (Counting Sort)		
m	n	Real Time(ms)
25	100000	150
	250000	650
	500000	1650
	750000	2700
	1000000	3826
50	100000	333
	250000	1467
	500000	3562
	750000	5793

Radix Sort (Counting Sort)		
	1000000	8166
75	100000	590
	250000	2417
	500000	5655
	750000	9165
	1000000	12837



Conclusion : From the above table, we can conclude that radix sort with the counting sort runs faster than radix sort with insertion sort.