YUQIAN ZHANG # N19945956 Recitation: Ojas

Fundamental Algoriths ProblemSet 3

Q1. Write each of follawing functions as $\Theta(g(n))$, where $g(n)$ is one of the standrad forms:

$2n^4 - 11n + 98 \sim \Theta(n^4)$

$6n + 43n\lg n \sim \Theta(n\lg n)$

$63n^2 + 14n\lg^5 n \sim \Theta(n^2)$

$3 + \frac{5}{n} \sim \Theta(1) \quad \because \lim_{n\to\infty} \frac{5}{n} = 0$

Q2 Illustrate the operation of RADIX-SORT on the list: COW, DOG SEA, RUG, ROW, MOB, BOX, TAB, BAR, EAR, TAR, DIG, BIG, TEA NOW, FOX (Alphabetical order and sort one letter at a time)

they are all 3 letters words, thus, 3 for loop needed in total

1st round loop: (Apply counting sort to 3rd letter)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
SEA	TEA	MOB	TAB	DOG	RUG	DIG	BIG	BAR	EAR	TAR	COW	ROW	NOW	BOX	FOX

2nd round loop: (Apply counting sort to 2nd letter base on result in 1st loop)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
TAB	BAR	EAR	TAR	SEA	TEA	DIG	BIG	MOB	DOG	COW	ROW	NOW	BOX	FOX	RUG

3rd round loop: (Apply counting sort to 1st letter base on result in 2nd loop)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
BAR	BIG	BOX	COW	DIG	DOG	EAR	FOX	MOB	NOW	ROW	RUG	SEA	TAB	TAR	TEA

Thus, above is the operation of RADIX-SORT of the list

Q3. Illustrate the operation of Bucket-SORT (with 10 buckets) on the array

A = (.79, .13, .16, .64, .39, .20, .89, .53, .71, .43)

There are 10 elements in the array A ⇒ N = 10

$\boxed{\text{step1}}$ For I = 1 to N

place = $\lfloor A[I] * N \rfloor$

insert A[I] into list B [place]

where list B is initialized with size 10 for 10 buckets

step1:

	A		⌊place⌋			List B		step2:	
1	0.79	→	7		0			0	
2	0.13	→	1		1	0.13 → $\boxed{0.16}$		1	0.13 → $\boxed{0.16}$
3	0.16	→	1		2	0.20		2	0.20
4	0.64	→	6		3	0.39		3	0.39
5	0.39	→	3	⇒	4	0.43		4	0.43
6	0.20	→	2		5	0.53		5	0.53
7	0.89	→	8		6	0.64		6	0.64
8	0.53	→	5		7	0.79 → $\boxed{0.71}$		7	0.71 → $\boxed{0.79}$
9	0.71	→	7		8	0.89		8	0.89
10	0.43	→	4		9			9	

$\boxed{\text{step2}}$ For J = 0 to N-1

sort B[J] By any sort

Thus, with the operation of Bucket-Sort showing above we have sorted Array B

B = (.13, .16, .20, .39, .43, .53, .64, .71, .79, .89)

Q4. Given $A[1 \cdots N]$ with $0 \le A[I] < N^N$ for all I.

(a) How long will COUNTING-SORT take?

The time complexity for COUNTING-SORT is $O\max(N, k)$

where k is the maximum number in the set,

N is the number of elements in the Array

Here, because $0 \le A[I] < N^N$ for all I in the Array A.

$k = N^N$

Then $O\max(N, k) = O\max(N, N^N) = O(N^N)$

Thus, COUNTING-SORT will take $O(N^N)$

(b) How long will RADIX-SORT take using base N?

The time complexity for RADIX-SORT is $O(D \cdot \max(N, K))$

where K is the base. D is digits in total

N is the number of elements in the set.

Here, $k = N$

To calculate how many digits we have for each element, we have

$D = \log_N(N^N) = N$

Then $O(D \cdot \max(N, k)) = O(N \cdot N) = O(N^2)$

Thus, RADIX-SORT will take $O(N^2)$ using base N

(c) How long will RADIX-SORT take using base $N^{\sqrt{N}}$ ($\sqrt{N}$ integral)

Base: $k = N^{\sqrt{N}}$

Digits: $D = \log_{N^{\sqrt{N}}}(N^N) = \sqrt{N}$

Then $O(D \cdot \max(N, k)) = O(\sqrt{N} \cdot \max(N, N^{\sqrt{N}}))$

$= O(\sqrt{N} \cdot N^{\sqrt{N}}) = O(N^{\sqrt{N}+\frac{1}{2}})$

By assumption, $\sqrt{N}$ can be seen as integral

Thus, RADIX-SORT will take $O(N^{\sqrt{N}+\frac{1}{2}})$ using base $N^{\sqrt{N}}$

Q5. Write the time T(N)

(a) $X = 0$

For $I = 1$ to N → Outter loop takes $\theta(N)$ time

do For $J = 1$ to N → inner loop takes $\theta(N)$ time

$x{+}{+} \rightarrow \theta(1)$

$\theta(N^2)$

(b) $I = 1$

WHILE $I < N$

do $I = 2 * I$

} → range is $1 \sim N$

increment by multiplication of 2

time $\log_2 N$

$\theta(\lg N)$

(c) For $I = 1$ to N

do $J = 1$

while $J * J < I$

do $J{+}{+}$

$\theta(N^{\frac{3}{2}})$

$$\begin{array}{cccccccc} & I=1 & 2 & 3 & 4 & 5 & \cdot & N \end{array}$$

$T(N) = 0 + \lfloor\sqrt{1}\rfloor + \lfloor\sqrt{2}\rfloor + \lfloor\sqrt{3}\rfloor + \lfloor\sqrt{4}\rfloor + \cdots \lfloor\sqrt{N-1}\rfloor \le N\sqrt{N-1}$

$\geq \frac{N}{2}\sqrt{\frac{N}{2}-1}$

$\therefore T(N) = \theta(N^{\frac{3}{2}})$

(d) For $I = 1$ to N

$J = I$

WHILE $J < N$

do $J = 2 * J$

$\theta(N)$

Inner loop:

$\log \frac{N}{j}$

Outter loop:

$T(N) = \sum_{i=1}^{N} \log \frac{N}{i}$

$= (\log \frac{N}{1}) + (\log \frac{N}{2}) + (\log \frac{N}{3}) + \cdots (\log \frac{N}{N})$

$= N \log N - (\log 1 + \log 2 + \log 3 + \cdots \log N)$

$= N \log N - \log(N!)$ ①

According to sterring series:

$\log(N!) \sim N \log N - n + 1$

Apply to formula ①:

$T(N) = N \log N - (N \log N - N + 1)$

$= N - 1$

Q6. Pro. Squander — BucketSort on n items with n^2 buckets
Ima Hogg — BucketSort on n items with $\sqrt{n}$ buckets
(Assume Ima's Algo for sorting inside a bucket takes $O(m^2)$ when bucket has m items)

(a) Argue that Prof. Squander has made a poor choice of the number of buckets.

By BUCKET-SORT Algorithm,
In first step, we must go over all buckets to put the data from original Array at once.
And that time depends on how many buckets we have
In Prof. Squander's case, he set up n^2 buckets for n items
So, the algorithm force it to run (n^2) time to go over all buckets
→ Best / Avg case: $\Theta(n^2)$ n items in distinct buckets
→ Worst case: $\Theta(n^2 + n^2)$ n items in 1 bucket
Thus, his algorithm cannot improve the run time for best and average case. At least n^2 time taken for BUCKET-SORT

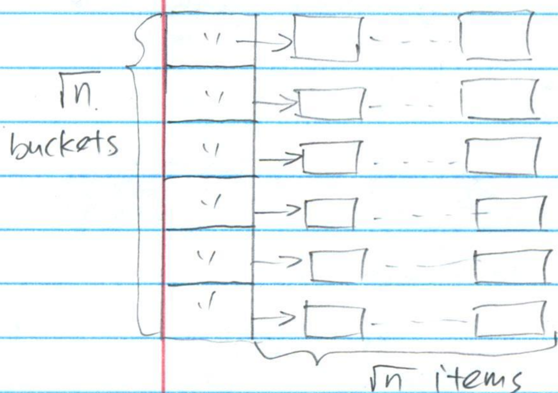
(b) Argue that Ima has a poor choice of # buckets

In Ima's case, she set up $\sqrt{n}$ buckets for n items
So, the algorithm force it to run $(\sqrt{n})$ time to go over all buckets
→ Best / Avg case: $\Theta(\sqrt{n} + n)$
[$\sqrt{n}$ buckets, each bucket has $\sqrt{n}$ items]
Sorting time inside a bucket is $O(n)$
→ Worst case: $\Theta(\sqrt{n} + n^2)$
All n items in 1 bucket
sorting time inside a bucket is $O(n^2)$
Thus, her algorithm cannot improve the run time for best average and worst case with additional $\sqrt{n}$ time added

(c) Ima uses roughly the same amount of space as some one using n buckets.

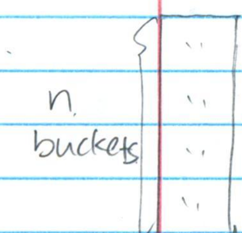
① $\boxed{\text{For the best / Avg case}}$

a) Ima's Algo: Each bucket has evenly ($\sqrt{n}$) items



Space usage $= \sqrt{n} \times \sqrt{n} = n$

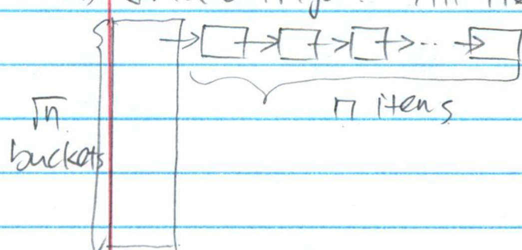
b) Some one's Algo: Each bucket has exactly 1 item



space usage $= n \times 1 = n$

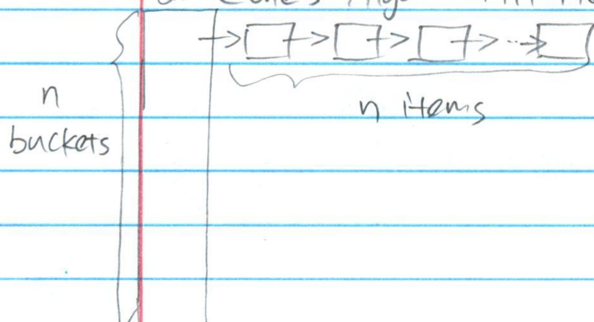
② $\boxed{\text{For worst case}}$

a) Ima's Algo: All items in 1 bucket



space usage $= \sqrt{n} + n - 1$

b) Some one's Algo: All items in 1 bucket.



space usage $= n + n - 1$

$\Rightarrow$ Thus, combine with Best/Avg and Worst, they roughly use same amount of space.