MD NOMAN AZAD

CS → 320

[redacted] 03/02/16

Problem #5 From HomeWork

$T(1) = 0$

$T(n) = 3T(n/3) + 2n - 3$

~~n = 3^1~~ $n = 3^k$

$k = \log_3 n$

$S(k) = T(n)$

$= 3S(3^k/3) + 2\cdot 3^k - 3$

$= 3S(3^{k-1}) + 2\cdot 3^k - 3$

$S(0) = T(1) = 0$

$$R(k) = \frac{S(k)}{3^k} = \left(\frac{3S(3^{k-1})}{3^k}\right) + \frac{2\cdot 3^k}{3^k} - \frac{3}{3^k}$$

$\Downarrow$

$R(k-1)$

$R(k) = R(k-1) + 2 - \frac{1}{3^{k-1}}$

$R(k) - R(k-1) = 2 - \frac{1}{3^{k-1}}$

⋮

$R(1) - R(0) = 2 - \frac{1}{3^0}$

$R(k)$ … $R(0)$: $2k - \sum_{i=0}^{k-1} \frac{1}{3^6}$

$$R(k) = 2k - \sum_{i=0}^{k-1} \frac{1}{3^6}$$

$$= 2k - \frac{\left(\frac{1}{3}\right)^k - 1}{\frac{1}{3} - 1}$$

$$= 2k - \frac{1 - \left(\frac{1}{3}\right)^k}{1 - \frac{1}{3}}$$

$$\cancel{= 2k - 1 - \frac{1}{3}}$$

$$= 2k - \frac{1 - \frac{1}{3^k}}{2/3}$$

$$= 2k - \frac{3}{2}\left[1 - \frac{1}{3^k}\right]$$

$$S(k) = 3^k R(k) = 2 \cdot k \cdot 3^k - \frac{3}{2} 3^k \left[1 - \frac{1}{3^k}\right]$$

$$= 2k\,3^k - \frac{3^k \cdot 3}{2} + \frac{3}{2}$$

$$T(n) = 2 \cdot \log_3 n \cdot 3^{\log_3 n} - \frac{3^{\log_3 n} \cdot 3}{2} + \frac{3}{2}$$

$$= 2n \log_3 n - \frac{3n}{2} + \frac{3}{2}$$

—— x ——

Average-Case Analysis of Quicksort

$$T(0) = T(1) = C$$

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

$$= Cn + \frac{2}{n} \sum_{k=0}^{n-1} T(k) = T(n * n)$$

$$= Cn^2 + 2 \sum_{k=0}^{n-1} T(k) \quad \text{——— } (1)$$

$$T(n+1) = C(n+1) + \frac{2}{n+1} \sum_{k=0}^{n} T(k) = T(n+1) \cdot (n+1)$$

$$\Rightarrow C(n+1)^2 + 2 \sum_{k=0}^{n} T(k) \quad \text{——— } (II)$$

After subtracting (I) from (II):

$$T(n+1)(n+1) - T(n)(n) = c(n+1)^2 - cn^2 + 2T(n)$$
$$= c[(n+1)^2 - n^2] + 2T(n)$$

$$\cancel{\frac{(n+1)T(n+1)}{(n+1)(n+2)} - \frac{(n)T(n)}{(n+1)(n+2)} = \frac{c(2n+1)+2T(n)}{(n+1)(n+2)}}$$

$$= c(2n+1) + 2T(n)$$

$$\frac{\cancel{(n+1)}T(n+1)}{\cancel{(n+1)}(n+2)} - \frac{\cancel{(n+2)}T(n)}{(n+1)\cancel{(n+2)}} = \frac{c(2n+1)}{(n+1)(n+2)}$$

$$c\,\frac{2n+1}{(n+1)(n+2)} = \frac{a}{n+1} + \frac{b}{n+2} = \frac{-1}{n+1} + \frac{3}{n+2}$$

$$a(n+2) + b(n+1) = 2n+1$$
$$(a+b)n = 2n$$
$$2a + b = 1$$
$$a = -1, \quad b = 3$$

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

$$R(n+1) - R(n) = c\left[\frac{-1}{n+1}, \frac{3}{n+2}\right]$$

$\vdots$

$$R(1) - R(0) = c\left[\frac{-1}{1}, \frac{3}{2}\right]$$

$$-c\sum_{i=1}^{n+1}\frac{i}{i} + c\sum_{i=1}^{n+1}\frac{3}{i+1}$$

$$\approx c_2 \ln n \approx R(n+1)$$

$$(n+1)\, 2\ln n \approx T(n+1)$$

Lower Bounds

Lower Bounds for minimum

→ The Lower bound defines the best possible efficiency for any algorithm that solves the Problem, including algorithms not yet invented.

→ A simple estimate for a problem's lower bound can be obtained by measuring the size of the input that must be read and output that must be written.

Lower Bounds for sorting

There are total $n!$ possible ways, among which 1 is Right only.

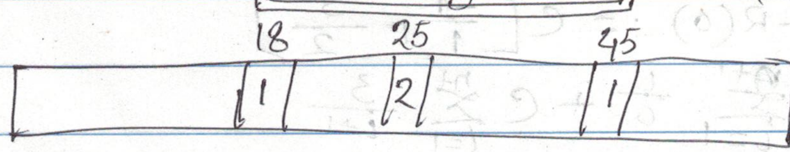
Q: If we have # leaves $\geq n!$, how big the tree needs to be?

Ans: $n \log n$

— x —

Counting Sort

(1 to m) (data)



25
45
18
25

a[k]++

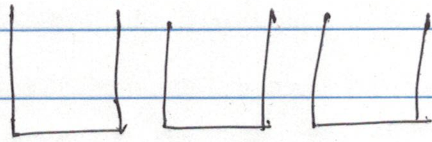
→ initialize m operations

→ Counting n

→ writeout m+n

$O(m+n)$

Bucket Sort



→ Arrange the data on ~~approp~~ appropriate buckets, based on specific characteristics. ~~The~~

→ Then sort the data in each bucket separetly.

$\frac{m}{k}$ → upper bound of Range

k → Number of Buckets

$\frac{n}{k}$ in each bucket

$$\text{Average} = \frac{\left(\frac{n}{k}\right)^2 k}{4} = \frac{n^2}{4k}$$

$$\Rightarrow k = \frac{n}{d}$$

<u>Worst Case</u>: One bucket gets all the Data.

— x —

Radix Sort

① 050, 150, 300, 006 } First, sorted by last digits

② 300, 006, 050, 150 } then, Sorted by second last digits

③ 006, 050, 150, 300 } then, sorted by third last digits (First digits).

Final sorted

data

0	5	0
0	0	6
1	5	0
3	0	0

Sort by last ~~digits~~ digits