Made with Goodnotes

Q2] a] $T(n) = 4T(n/2) + n^{\log_4 8}$

$n^{\log_b a} = n^{\log_2 4} = n^2$

$n^{\log_4 8} = n^{1.5}$

$\therefore$ case 1 : $T(n) = \Theta(n^2)$ $\quad \epsilon = 0.5$

b] $T(n) \le ck^{2.5}$

$\Rightarrow 4c\left[\frac{n}{2}\right]^{2.5} + 4n^{2.5} \le cn^{2.5}$

$\Rightarrow \cancel{n^{2.5}}\left[\frac{4c}{2^{2.5}} + 4\right] \le c\cancel{n^{2.5}}$

$\Rightarrow \frac{c}{\sqrt{2}} + 4 \le c$

$\Rightarrow 4 \le c - \frac{c}{\sqrt{2}}$

$\Rightarrow 4 \le \frac{c\sqrt{2} - c}{\sqrt{2}}$

$\Rightarrow \frac{4\sqrt{2}}{\sqrt{2}-1} \le c$

Made with Goodnotes

LO3] a] When attempting to reduce the array size by 25% via random pivot selection, followed by partitioning, the worst case occurs when the desired lies in the interval $[n/4, 3n/4]$. In this case, the pivot must also be selected from this interval which has a probability of 0.5. Thus $1/0.5 = 2$. It is the expected no. of pivot selections needed to reduce the array by 25% & since a constant no. of pivot selections & partition steps require $O(n)$ steps we get the reccurrence $T(n) = T(3n/4) + n$.

b] $x = 10100101$ $x_L = 1010$ $x_R = 0101$

$y = 11110000$ $y_L = 1111$ $y_R = 0000$

$xy = 165 \times 240$

$= 39600$

$P_1 = 10010110 = 150$

$P_2 = 00000000 = 0$

$P_3 = 11100001 = 225$

$xy = P_1 \times 2^n + (P_3 - P_1 - P_2) \times 2^{n/2} + P_2$

$= 150 \times 2^8 + (225 - 150 - 0) \times 2^4 + 0$

$= \underline{39600}$

Made with Goodnotes

LO4] a] n being a positive power of 2 makes it constantly even. Then ω_n^i & $-\omega_n^j$ are both roots of unity. Furthermore, squares of the n^{th} roots of unity yield the $n/2$ roots of unity & so problem size gets halved. Thereby when evaluating these roots, no. of steps are $O(n \log n)$

b] $DFT_4^{-1}(5, 2, 3, -4)$

$DFT_2^{-1}(5, 3)$ odd

$DFT_2^{-1}(2, -4)$ even

$\Rightarrow DFT_2^{-1}(5,3) = (5,5) + (3,-3)$

$= \frac{1}{2}(8,2)$

$= (4,1)$

$= (4,1,4,1)$

$DFT_2^{-1}(2,-4) = (2,2) + (-4,4)$

$= \frac{1}{2}(-2,6)$

$(-1,3)$

$(-1,3,-1,3)$

$\Rightarrow (4,1,4,1) + (-1(1), 3(-i), -1(-1), 3(i))$

$\Rightarrow (4-1, 1-3i, 4+1, 1+3i)$

$\Rightarrow (3, 1-3i, 5, 1+3i)$

Made with Goodnotes

Q5) a] i] An i-neighboring path from s to v that uses exactly one edge not in D_i

ii] $d_i(s,v)$ is minimum cost of any i-neighboring path from s to v

b] The greedy choice is to add the vertex v to DDT_i that has the minimum cost of any i-neighboring path from s to v

c] Let P be the i-neighboring path from s to v^* where $cost(P) = d_i(s, v^*)$. Let R be another path from s to v^* where $s \ldots u \in DDT_i$ & $v \notin DDT_i$. Now vertex v has to exist if $v^* \notin DDT_i$. So $Q = s, \ldots u, v$ is an i-neighboring path & since P is minimum cost of all paths & Q is a subpath of R then $cost(R) > cost(Q) > cost(P)$

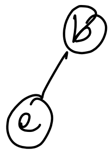
$\therefore d_i(s,v) = d(s,v)$.

Made with Goodnotes

LOG) step1 : b, e, 1

find(b) = b

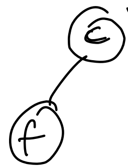
find(e) = e



step 2: c, f, 2

find(c) = c

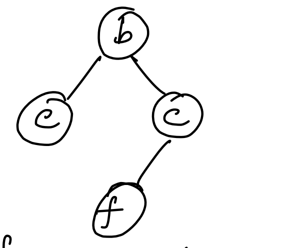
find(f) = f



step3: e, f, 3

find(e) = b

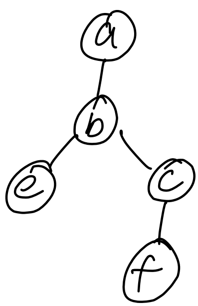
find(f) = c



step 4: a, e, 4

find(a) = a

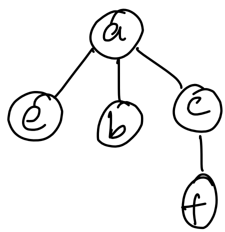
find(e) = b



step5: c, e, 5

find(c) = a

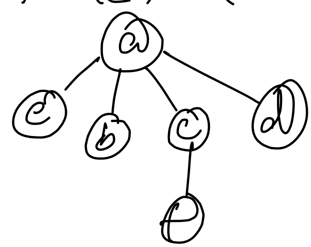
find(e) = a



step: d, e, 6

find(d) = d

find(e) = a



Made with Goodnotes