Q1] Refer to LCS from previous week soln.

Q2] a] Minimum no. of edit operations needed to convert one word to another.

b]

$$d(i,j) = \begin{cases} i & \text{if } i = 0 \\ j & \text{if } j = 0 \\ \min(d(i-1,j)+1,\ d(i,j-1)+1,\ d(i-1,j-1) + c(u_i \neq v_j)) \end{cases}$$

c] $u = cbbcca \quad v = bccaba$.

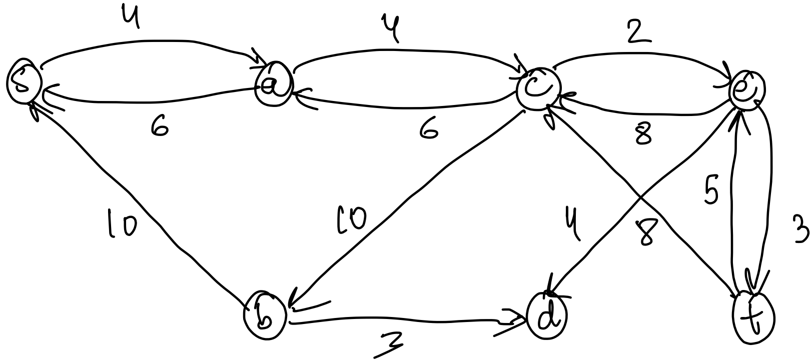
	λ	b	c	c	a	b	a
λ	0	1	2	3	4	5	6
c	1	1	1	2	3	4	5
b	2	1	2	2	3	3	4
b	3	2	2	3	3	4	4
c	4	3	2	3	4	4	5
c	5	4	3	2	3	4	5
a	6	5	4	3	2	3	4

∴ Total edits from cbbcca to bccaba are 4

→ delete c ⇒ bbcca ; delete b ⇒ bcca.

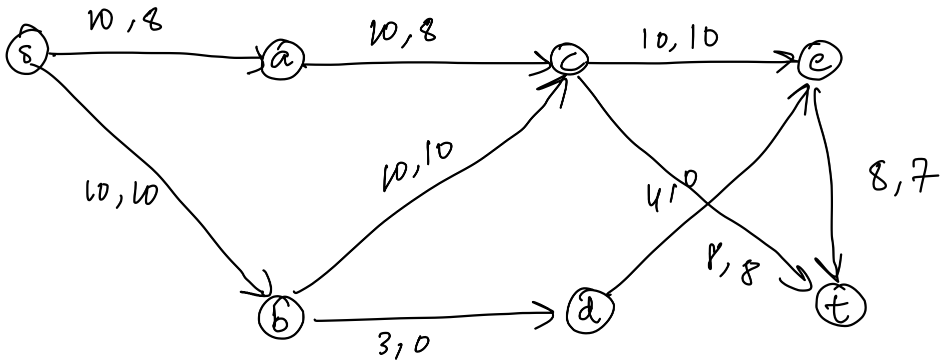
→ add b ⇒ b ; add a ⇒ bccaba

Q3] a]



min = 2

b]



c] reachable (G_f, s, t)

$G_f \Rightarrow$ graph network

$s \Rightarrow$ origin of path P.

$t \Rightarrow$ terminatory point of path P.

Q4] a] If s opt does not have a task presumably a' then it would mean that the time slot k. This will then result in Sopt having less no. of tasks than greedy soln. This is a contradiction as Sopt should have the max no. of tasks possible. If Sopt had no task at tk then we could add task a at any time & have a more profitable solution which contradicts the

Made with Goodnotes

b] The greedy algorithm's purpose is to make a solution with maximized profit. At a given time t_k, the algorithm can pick from a list of tasks in decreasing order of achieveable profit. At this point, the algorithm will pick the most profitable task given its deadline fits. For a replacement to occur we cannot decrease the profit of the task as it will keep going down further ahead. Therefore for a replacement to occur profit of a & a' should necessarily be equal or if possible $p \geq p'$. When the algorithm reaches time slot K, since neither a or a' have yet to be scheduled, both will appear in L_k. Moreover since the algorithm selects a, it follows that $p \geq p'$ & so replacing a' with a in Sopt results in another optimal solution which agrees with S from m down to k.

5] a]

$$P(i) = \begin{cases} 0 & \text{if } i = 0 \\ \min\limits_{1 \le k \le i} \left(penalty(k, i) + P(k-1) \right) & \text{otherwise} \end{cases}$$

$$penalty(k, i) = \begin{cases} \infty & \text{if } \sum_{j=k}^{i} w_j > M \\ \left(M - \sum_{j=k}^{i} w_j\right)^2 & \text{otherwise} \end{cases}$$

b]

i	0	1	2	3	4	5	6	7
P(i)	0	49	16	52	25	50	36	26

6] a] Proof of correctness of Prim's is similar to Kruskal's. In Kruskal's, edges are sorted according to their weights. In this manner, the negative edges will be handled by the union find data structure. Negative weight edges that don't lead to a cycle will be considered as they can be advantageous to the MST. Being able to detect a cycle formation in this case helps Prim's handle negative edge weights. There is no requirement that edges have nonnegative weight. All that matters is that the edges have an ordering based on respective weights. No step of the proof uses $W >= 0$

Made with Goodnotes

b) Let $w < 0$ have the least weight of any edge of G. Then Eswar should add $[w]$ to each edge of G so that the least weight will now equal $w + [w] = 0$ & he may now use the program. Now suppose the program returns an MST T with cost C. Then the actual cost for T based on G's original weights is $Cost(C) = (e-1)[w]$ since $[w]$ must be subtracted from each edges $(n-1)$ of T.

LO Makeup.

LO1] a] $86 - 37(2) = 12$

$37 - 12(3) = 1$

$37 - 12(3) = 1$

$37 - (86 - 37(2))(3) = 1$

$37 - 86(3) + 37(6) = 1$

$37(7) - 86(3) = 1$

$\therefore$ multiplicative inverse $\underline{= 7}$.

b] $a^{\frac{n-1}{2}} = \left(\frac{a}{n}\right) \bmod n$

$\Rightarrow 2^2 = \frac{2}{5} \bmod 5$

$\Rightarrow 4 \bmod 5 \equiv -1$

$\therefore LHS = -1$

$\frac{2}{5} \bmod 5$

$-1 \equiv -1 \bmod 5$

$\therefore LHS = RHS$ $\therefore$ a is an accomplice for n=5 being prime.

LO2) a] $n^{\log_b a} = n^{\log_4 16} = n^2$

$f(n) = n^{\log_3 9} = n^2$

$\therefore n^{\log_b a} = \Theta(f(n))$

$\therefore T(n) = \Theta(n^2 \log n)$

b] $T(n) = O(n^{1.5})$

$\Rightarrow T(k) \le ck^{1.5}$ for $k < n$

$\therefore 2c\left(\frac{n}{2}\right)^{1.5} + n^{1.5} \le cn^{1.5}$

$\Rightarrow \frac{2c}{2^{1.5}} + 1 \le c$

$\frac{c}{\sqrt{2}} + 1 \le c$

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

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

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

$\Rightarrow \sqrt{2} \le c(\sqrt{2}-1)$

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

LO3] a] $r = ae + bg \Rightarrow P_5 + P_6 - P_2 + P_4$

$s = af + bh \Rightarrow P_1 + P_2$

$t = ce + dg \Rightarrow P_3 + P_4$

$u = cf + dh \Rightarrow -P_7 + P_5 + P_1 - P_3$

b] 5 4 12 8 7   1 13 9 10 16

5 4   12 8   7   1 13   9 10   16

4 5   8 12   7   1 13   9 10   16

4 5   7 8 12   1 13   9 10 16

4 5 7 8 12   1 9 10 13 16

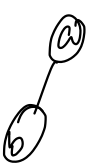
1 4 5 7 8 9 10 12 13 16

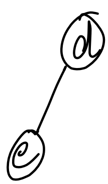
LO4] a) & b] refer to LO7 10-18-2023.

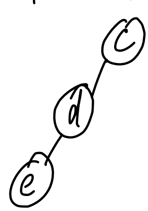
LO6] step1 ⇒ a, b, 1   step2 ⇒ e, d, 2   step3 ⇒ e, c, 3

find (e) ⇒ d

find (c) = c



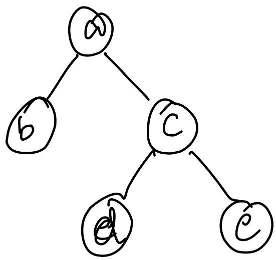




step 4 ⇒ a, e, 4

find(a) = a

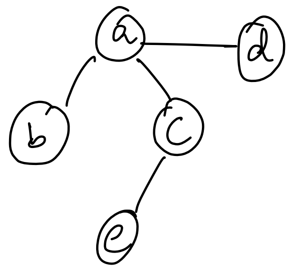
find(e) = e



step 5 ⇒ b, d 5

find(b) = a

find(d) = a



step 6 ⇒ e, f, 6

find(e) = a

find(f) = f