1) To prove that E can be decomposed into K matchings we use the concept of Contradiction.

→ We initially prove that if $G(U, W, E)$ is a k-regular bipartite graph then E can be decomposed into k-matchings.

$G(U, W, E) \leftarrow$ k-regular bipartite graph
$(V_1, V_2) \leftarrow$ Bipartition of $G(U, W, E)$.

→ Since G is k-regular we get

$k|V_1| = |E| = k|V_2|$

$\Rightarrow |V_1| = |V_2|$ [because $k > 0$]

Based on the assumption that W is a subset of V_1 and the edges D_1 and D_2 correspond to vertices in W and $T(W)$ we have

$D_1 \subseteq D_2$ [From $T(W)$]

$\Rightarrow k|T(W)| = |D_2| \geq |D_1| = k|W|$

$\Rightarrow k|W| \leq k|T(W)|$

$\Rightarrow |W| \leq |T(W)|$

From the above results we can decipher that M is a perfect matching. Which further implies that graph $G(U, W, E)$ has a perfect matching.

In the question, it is given that $G(U, W, E)$ is a bipartite graph with maximal vertex degree k.

→ If we remove all vertices then the graph will have a degree of 0. We can use this to find perfect matching.

→ We can do this by repeatedly removing the smaller side.

→ The graph will have $(n-k)$ edges after the above step where
$n \leftarrow$ total no. of edges in the ~~ste~~ graph
$k \leftarrow$ no. of steps carried out.

The remaining edges cannot be a part of any previous matchings found because otherwise atleast one of its end points would have an edge in all the matchings

→ We can hence say that there exists a vertex in graph $G(U, W, E)$ which has a degree greater than k.

Hence Proved that E can be decomposed into k matchings because the above is a proof of contradiction.

2.) To prove that :

$$|N(S)| + |N(T)| \geq |N(S \cap T)| + |N(S \cup T)|$$

Given :

→ $G = (U, W, E)$ be a bipartite graph

→ $S, T \subseteq U$

→ $N(S \cup T) = \{a = w \mid ab \in E$ for atleast one $b \in S \cup T\}$

[neighbour of $S \cup T$].

→ $N(S \cup T) \subseteq N(S) \cup N(T)$

→ $N(S \cup T) \leq |N(S)| + |N(T)|$

~~The elements in $N(S \cup T)$~~

The elements in $N(S \cap T)$ are counted only once on left hand side whereas they are counted twice on right hand side as $N(S \cap T) \subseteq N(S)$ and $N(S \cap T) \subseteq N(T)$.

This gives us → $|N(S \cap T)| + |N(S \cup T)| \leq |N(S)| + |N(T)|$

Hence Proved

3.) $G = (U, W, E) \leftarrow$ Bipartite graph.
$M \leftarrow$ Maximum matching of G.

We need to give a linear time algorithm, satisfying the given condition

$$|N(T)| - |T| = \min_{S \subseteq U} (|N(S)| - |S|)$$

We know that size of M cannot exceed $(|U| - |S| + |N(S)|$
[For any matching there should be at most $|N(S)|$ incident edges]
[The maximum no. of incident edges to U/S can be $|U| - |S|$.]

$S = U/C$
where $C \leftarrow$ maximum vertex cover.

From the above, we can conclude that $N(S) \in W \cap C$.
This gives us $\rightarrow$

$$|S| - |N(S)| \geq |S| - |W \cap C| = |U/C| - |W \cap C| = |U| - |U \cap C| - |W \cap C| = |U| - |C|$$

$$\Rightarrow |U| - |S| + |N(S)| \leq |C|$$

$$\Rightarrow T(G) = \min |C| = \min_{S \subseteq U} (|U| - |S| + |N(S)|$$

$$\Rightarrow \alpha'(G) = T(G) = \min_{S \subseteq U} (|U| - |S| + |N(S)|)$$

(Using Konig's Algorithm).

4.) Given $S, T \in \mathcal{T}$ and $|S| < |T|$

We know that in a bipartite graph, $x \subseteq y$ if no two edges of A are identical or have the same vertex.

$\therefore$ we get $\rightarrow (x, y)$ $(V = x \cup y)$

This gives us $L = \{ A \subseteq x$: there exists a matching m and all vertices of A are matched $\}$.

$\Rightarrow B \in L$ because $A \in L$ and $B \subseteq L$
Also, we have $B \in B/A$ so that $A \cup \{B\} \in L$.

Now we can use the following equations $\Rightarrow$

$A, B \in L$ and $|A| < |B|$ and
$B \in L$ as $A \in L$ and $B \subseteq L$

$\therefore$ There exists ~~$v \in V \setminus S$~~ $v \in T \setminus S$ such that $S \cup \{v\} \in \mathcal{T}$

Hence Proved

5.) We will use the Hungarian Algorithm.

$G = (V, E_L) \leftarrow$ Equality graph
$u \leftarrow$ vertex
$M \leftarrow$ initial Matching
$L \leftarrow$ labelling

→ $\forall x \in X, y \in Y \mid L(Y) = 0$
$L(x) = Max_{y \in Y} (w(x, y))$.
Start with a random matching M and labelling L.

→ Repeat: until (M == Perfect):

a) Find an augmenting path in M.

b) if (path does not exist)
→ improve labelling & repeat step (a).

The above is the basic algorithm:

→ Now we have the algorithm for steps (a) and (b) as follows:

(a): Finding an augmenting path in M:

* Note augmenting path starting a u.
We consider a path based on the following
→ It should alternate between the edges in the matching
→ The 1st and last vertices are free vertices.

→ The edges should not be in the matching.

* If (an unmatched vertex is found): append it to the existing augmenting path p.
Add u to v segment.

* Replace the edges in M with edges in p which are not present in M.
By doing this we flip the matching.

(b): Improving the labelling:

$S, T \leftarrow$ candidate augmenting alternating path
$T \subseteq Y$ and ~~$x \in$~~ $S \subseteq X$
$N_L(S) \leftarrow$ Neighbours of node in S along E_L.

$N_L(S) = \{ v \mid \forall u \in S : (u, v) \in E_L \}$.

→ We know we cannot improve the labelling if the size of the alternating path is not increasing anymore.

→ $\delta_L \leftarrow$ minimimm of $l(u) + l(v) - W(u, v)$ where $u \in S$ & $v \in T$.

Inorder to obtain the improved labelling

$l'(r) = l(r) - \delta_L$ (if $r \in S$)

$$L'(r) = L(r) + \delta_L \quad (\text{if } r \in T)$$
$$L'(r) = L(r) \quad (\text{if } r \notin S \text{ \&\& } r \in T)$$

Therefore, L' is a valid labeling & $E_L \subset E_{L'}$.

6a.) $D = (V, A; l)$
where $l \leftarrow$ edge length function.

When a subset of edges cover all the vertices then it is considered a perfect matching. For a graph to have a perfect matching the number of vertices must be even.

$|A| = |V|$ and $l(e) = \begin{cases} w(e) & \text{if } e \in M \\ -w(e) & \text{otherwise} \end{cases}$

Therefore we can conclude that every vertex in graph is represented in M.

The edges present in the matching will be positive [based on the edge length function above].

If D has a circuit then it will be negative as it will be made up of the majority of the negative weighted edges resulted from the above $l(e)$ function. We get this from the fact that G has a perfect matching and $D = (V, A; l)$ has a circuit.

Therefore, D has a negative circuit if and only if G has a perfect matching with negative weight.

In the circuit D, the edges have 0 weight and the ~~ve~~ matching has negative weigh making the circuit negative.

Scanned with CamScanner

6 b) We know that the matching is perfect so M is also an edge cover.
path $s \to t$ will give us graph G' by removing edges s_{in} and s_{out}.

Using the edge length function only G' will have positive weight

$$l(e) = \begin{cases} W(e) & \text{if } e \in M \\ -W(e) & \text{otherwise} \end{cases}$$

[We get this because we know G' has a perfect matching].

D is a circuit of which graph G' is a part we get

$$\min_{e \in M} \{W(e)\} + \min_{e \notin M} \{W(e)\}.$$

Hence, using minimum weight perfect matching we can conclude that $e \in M$ will not include s_{in} and s_{out}.

Therefore, Hence Proved that the shortest path in circuit D has been derived from minimum weight perfect matching.