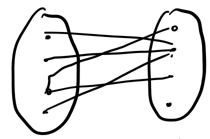
Q1. Bipartite graph :

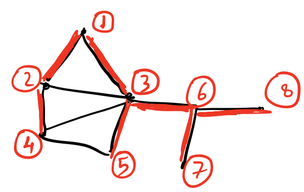


a) Suppose there is an odd length cycle containing nodes $v_1, v_2, v_3, \ldots, v_k$.

WLOG, say $v_1 \in X$
then $v_2 \in Y$
$v_3 \in X$
$\vdots$
$v_{2i+1} \in X$
$v_{2i} \in Y$.

Since k is odd, $v_k \in X$. But there is an edge from v_k to v_1 which violates the condition of a bipartite graph.

b) BFS tree :



Observe: for any edge $(a,b) \in G$
$|level(a) - level(b)| \leq 1$.

Part 1 : If $\exists$ edge $(a,b) \in E$ s.t. $|level(a) - level(b)| = 0$ in a BFS tree of G, then G is not bipartite.

Consider x = LowestCommonAncestor(a,b)
Consider the cycle formed by treepath(x,a) :: (a,b) :: treepath(b,x)
This is an odd cycle of length $2 * dist(x,a) + 1$.
$\therefore$ By part 1, G cannot be bipartite. $\square$

Part 2: If $\forall$ edge $(a,b) \in E$, $|level(a) - level(b)| = 1$ in any BFS tree of G, then G is bipartite.

Can put all odd-level nodes in set X and all even-level nodes in set Y.

No edge within nodes of set X or Y.

$\square$

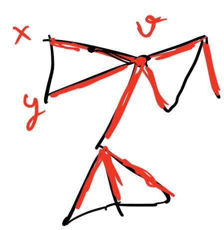
c) Consider a connected graph first.

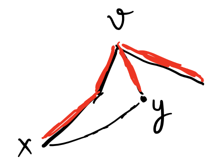
Construct BFSTree(G) — $O(n)$

$\forall$ edge $(a,b) \in$ BFSTree(G), check if $|level(a) - level(b)| = 1$ — $O(m)$

For unconnected graphs, repeat for each connected component and report 'bipartite' if all connected components are bipartite.

Q2. For any vertex v in G, let $T_v = BFS(v)$.
Let $e_v = (x,y)$ be a non-tree edge w.r.t. T_v
s.t. $L_v = Level(x, T_v) + Level(y, T_x)$ is minimum.





Algorithm (for finding shortest cycle):

- Let v be the vertex for which L_v is the least.
- Let $e_v = (x,y)$ be defined as above.
- set $C = treepath(v,x) :: (x,y) :: treepath(y,v)$ and output $|C|$.

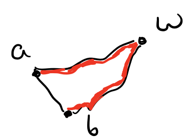
Claim 1: $|C| = 1 + L_v$.

<u>Pf:</u> $|C| = |treepath(v,x)| + treepath(y,v) + 1$
$= L_v + 1.$

$\square$

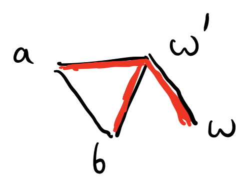
Claim 2: C is a min-length cycle.

Pf: Let (a,b) be a non-tree edge in min-length cycle c'.

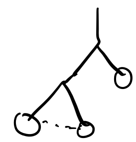


$$\text{Length of } c'[w,a] \geq dist(w,a)$$
$$\text{Length of } c'[w,b] \geq dist(w,b)$$
$$\Rightarrow |c'| \geq dist(w,a) + dist(w,b) + 1$$
$$\geq L_w + 1.$$

Also, LeastCommonAncestor(a,b) must be w, otherwise treepath(w', a) :: (a,b) :: treepath(b,w') would be a smaller cycle, where w' = LeastCommonAncestor(a,b).



Q3. (i) If x is a leaf node then vertices in $T \setminus \{x\}$ are connected. $\Rightarrow$ Vertices in $G \setminus \{x\}$ are connected.



(ii) Root of T is cut vertex $\Rightarrow$ Has at least two children

If single child, then root acts as leaf. Claim follows from part (i). $\square$

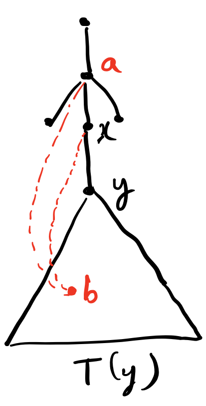
Root of T has at least two chidren $\Rightarrow$ Is cut vertex.

Suppose a_1 and a_2 be two children of x.
a_1 and a_2 are disconnected in $G \setminus \{x\}$ because there is no cross edge connecting subtrees $T(a_1)$ and $T(a_2)$.

(Recall : DFS tree has no cross edge). $\square$.

(iii) High point(y) : Level of the highest ancestor of y to which there is a back edge from descendants of y, if such a back edge exists.

If no such back edge exists, high point(y) = level(y).



To prove: x is cut vertex iff Highpoint(y) $\geq$ Level(x).

Claim 1: If highpoint(y) < level(x) then removing x does not disconnect $T(y)$.

Pf: a must be proper ancestor of x.
Thus, treepath(y, b) :: (b, a) :: treepath(a, root) connects $T(y)$ to the root. $\square$

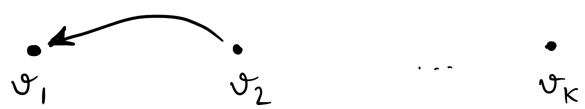
Claim 2: If highpoint(y) $\geq$ level(x) then removing x disconnects $T(y)$

Pf: If we remove x, then there is no path from the root of the DFS tree to $T(y)$ as all edges originating from $T(y)$ only reach vertices of $T(y)$ in $G \setminus \{x\}$.
(Recall: no cross edge in DFS tree) $\square$.

(iv) For each connected component of G, check conditions (i), (ii) and (iii).

Q4.

can reach from C_2 to C_1 but not from C_1 to C_2

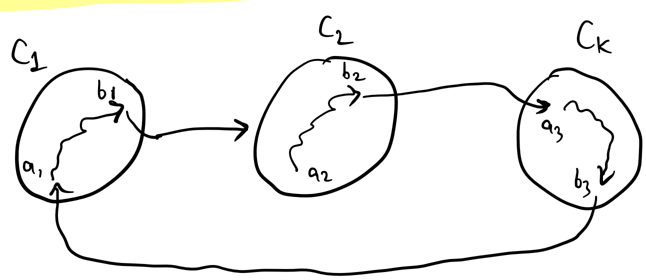


(i) Compute SCCs of G: Tarjan's algorithm

Iterate over edges in G, and put edges in H accordingly.

Time taken: $|E(G) + V(G)|$
$= m + n$

(ii) **Claim:** If there is a cycle in H, say $(v_1, \ldots, v_K)$ then vertices in $C_1 \cup \ldots \cup C_K$ must be strongly connected to each other.



Contradiction because strongly connected components are maximal. $\square$