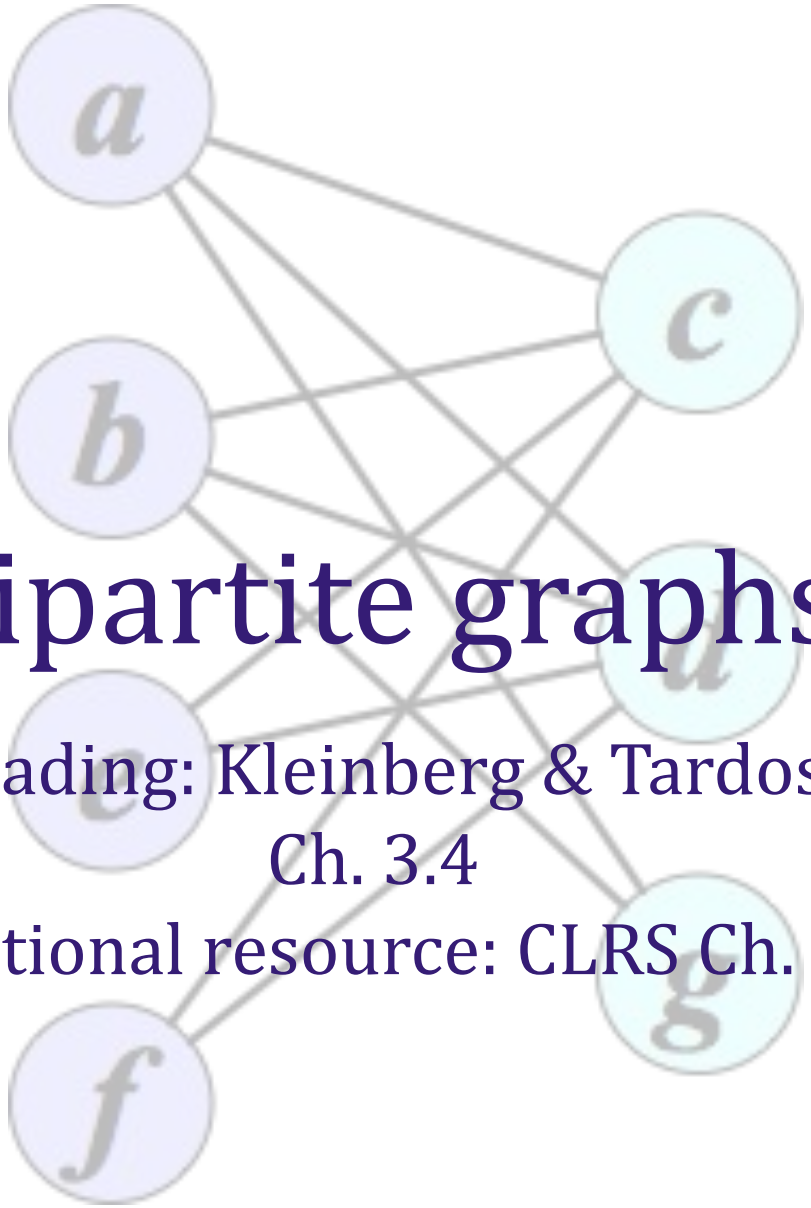


Bipartite graphs

Reading: Kleinberg & Tardos
Ch. 3.4
Additional resource: CLRS Ch. 22

Icebreaker

Suppose you are trying to organize a first-year orientation event, and your goal is to get students who don't already know each other to meet. (Students that already know each other will probably start clustering…)

Can you send the students into two different rooms so that each contains students who don't know anyone else there?

Thanks to Jessica Sidman for posing this problem!

Icebreaker

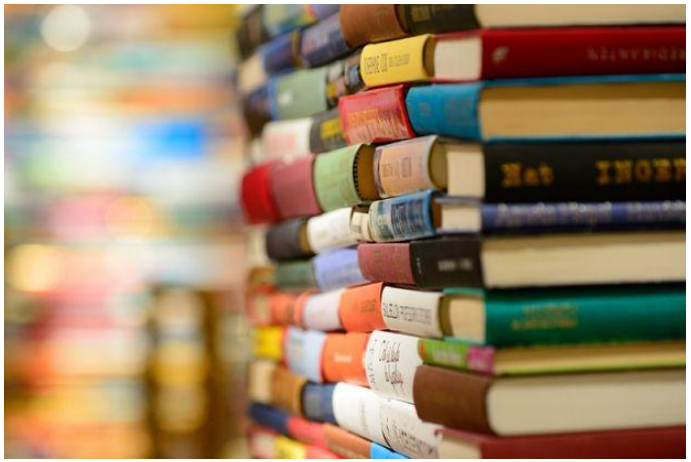
Construct (undirected) graph

- Vertex for each student
- Edge between two students if they know each other

Is it possible to divide the students into two groups so that neither group contains a pair of students who know each other?

Working together

Suppose you and your friend have a research project to study the poetry of Emily Dickinson. You've been given a list of books about her poems upon which to base the project. However, neither one of you wants to read analysis of the same poem more than once…

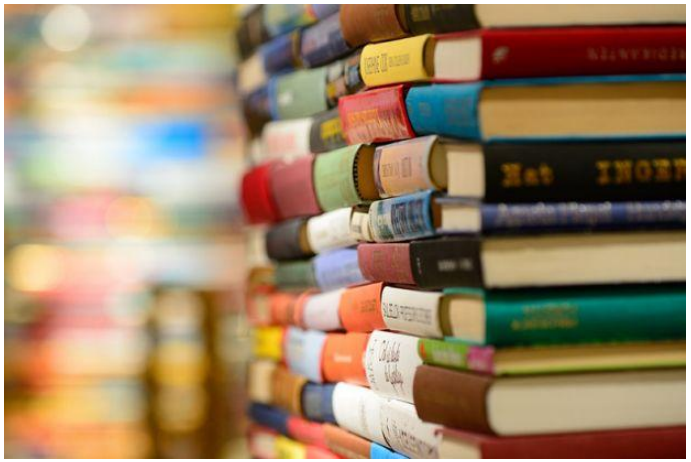


Is it possible to divide the books between you so that no one reads about the same poem more than once?

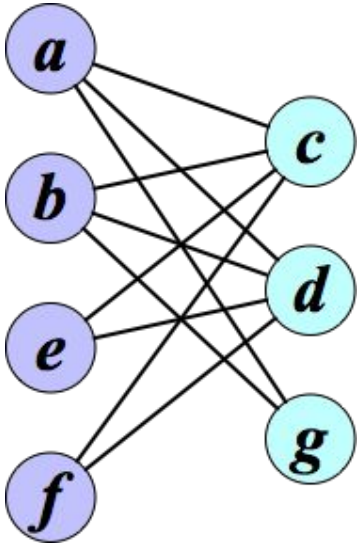
Working together

Construct (undirected) graph

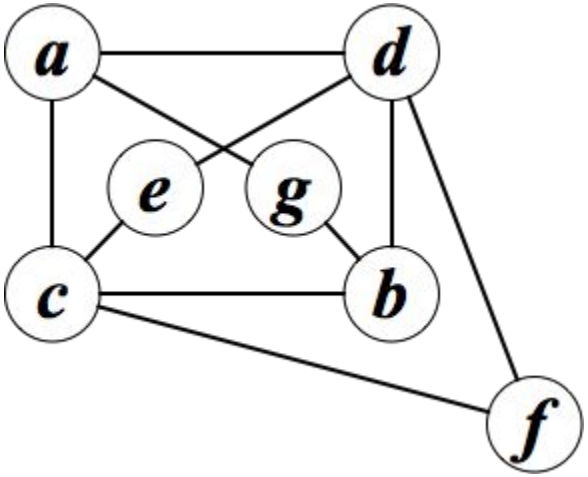
- Vertex for each book
- Edge between two books if analyze same poem

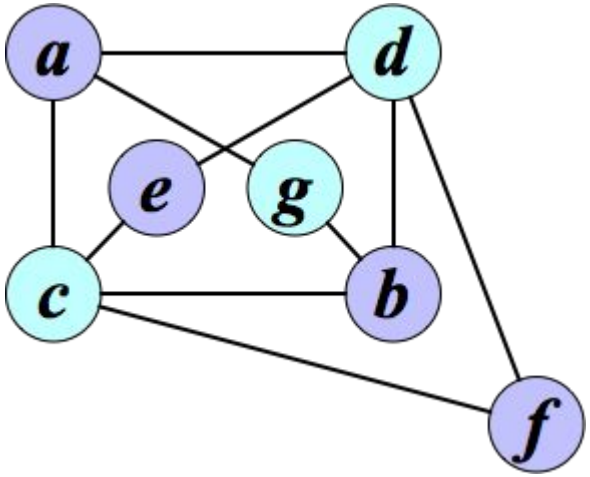


Is it possible to partition the vertices into two sets such that no one edge connects two vertices in the same set?



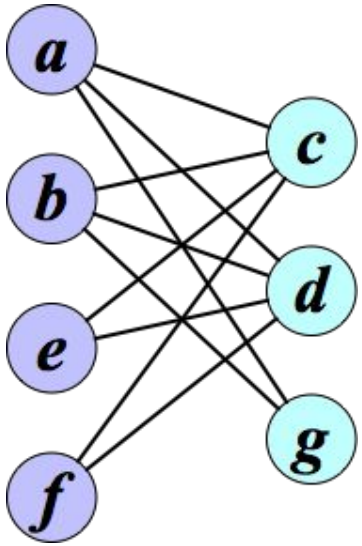
Bipartite graphs





Bipartite graphs

- Bipartite graph $G = (V, E)$
- V can be partitioned into disjoint vertex sets: X and Y
- Edges: $E \subseteq X \times Y$
  - One endpoint in each set



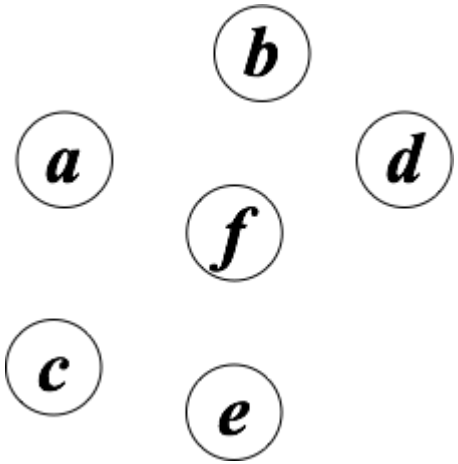
Example:

X = {a, b, e, f}, Y = {c, d, g}

E = {ac, ad, ag, bc, bd, bg, ec, ed, fc, fd}

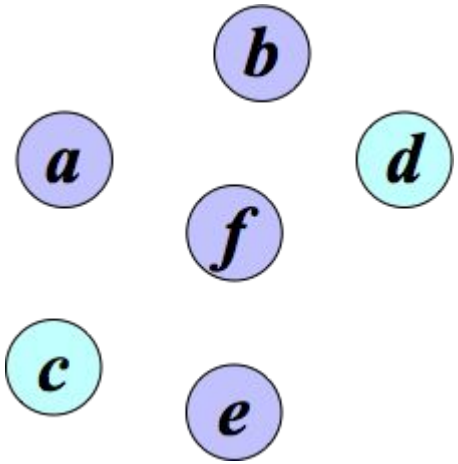
Is a graph bipartite?

Empty graph?



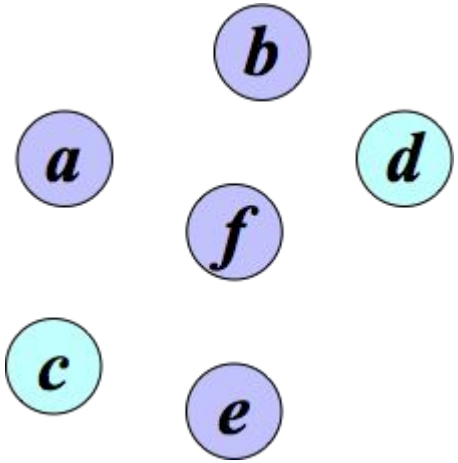
Is a graph bipartite?

Empty graph?

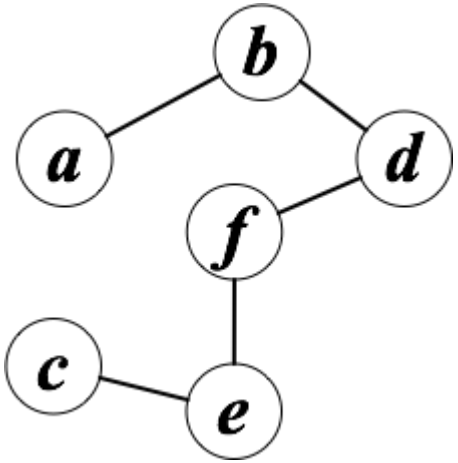


Is a graph bipartite?

Empty graph?

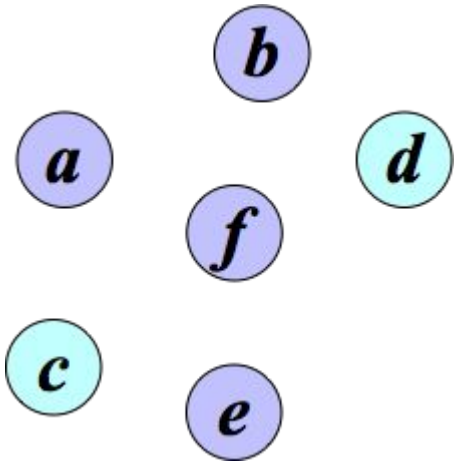


Path?

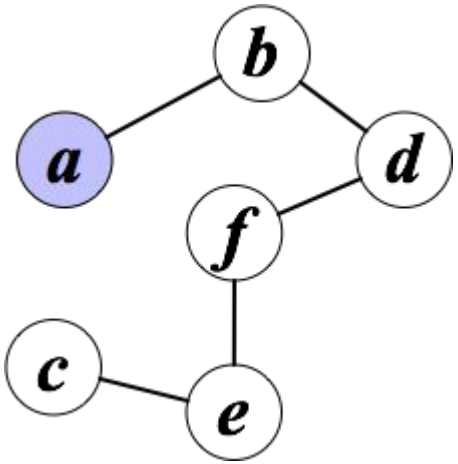


Is a graph bipartite?

Empty graph?

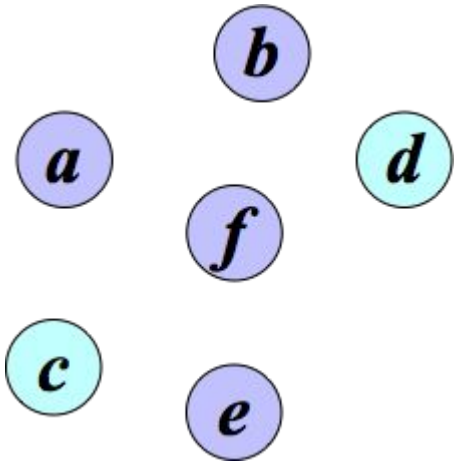


Path?

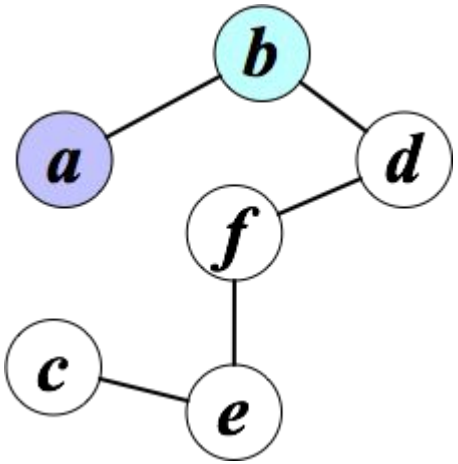


Is a graph bipartite?

Empty graph?

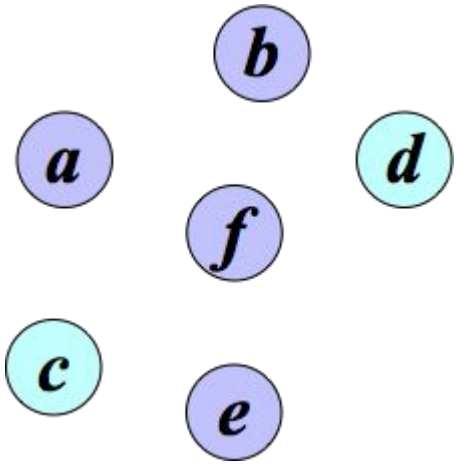


Path?

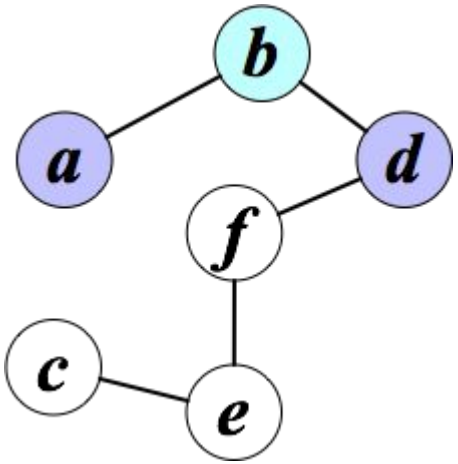


Is a graph bipartite?

Empty graph?

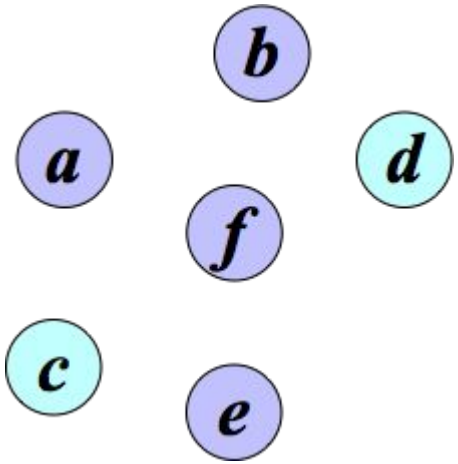


Path?

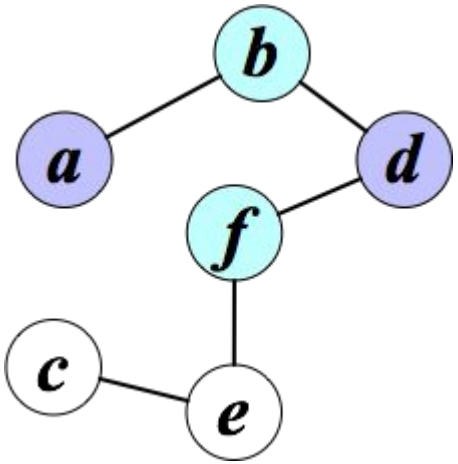


Is a graph bipartite?

Empty graph?

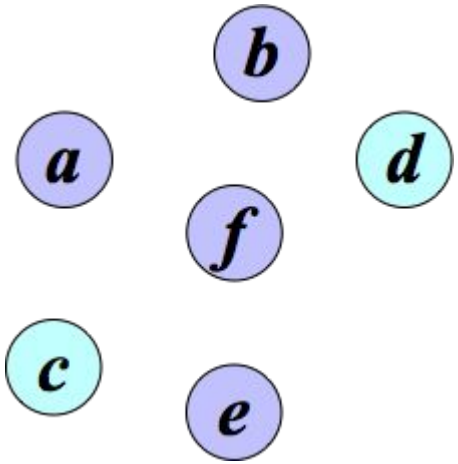


Path?

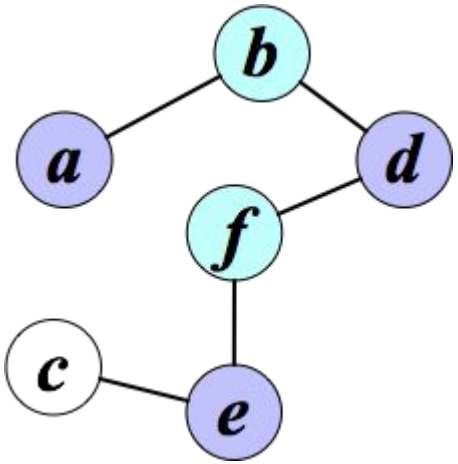


Is a graph bipartite?

Empty graph?

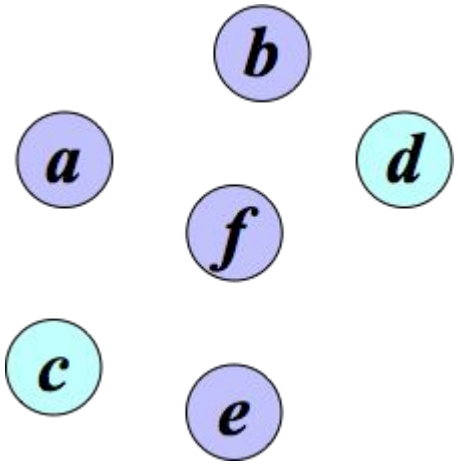


Path?

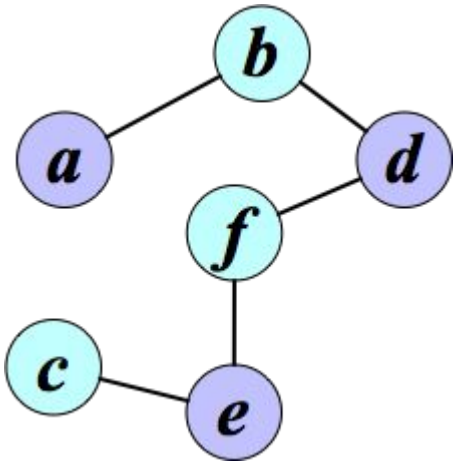


Is a graph bipartite?

Empty graph?

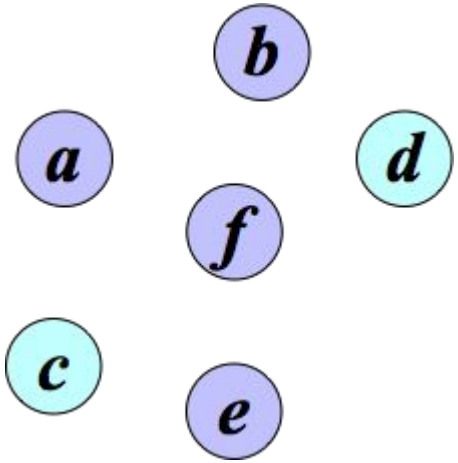


Path?

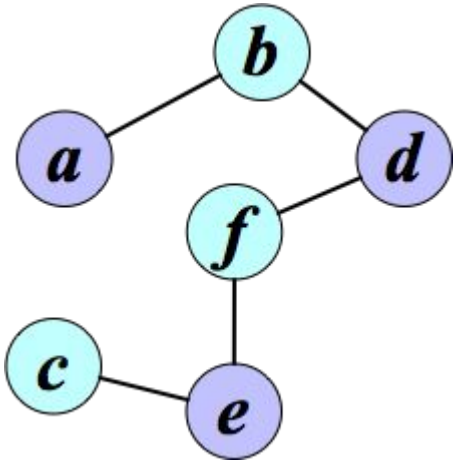


Is a graph bipartite?

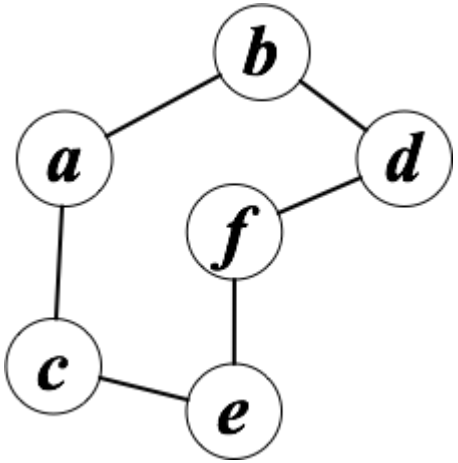
Empty graph?



Path?

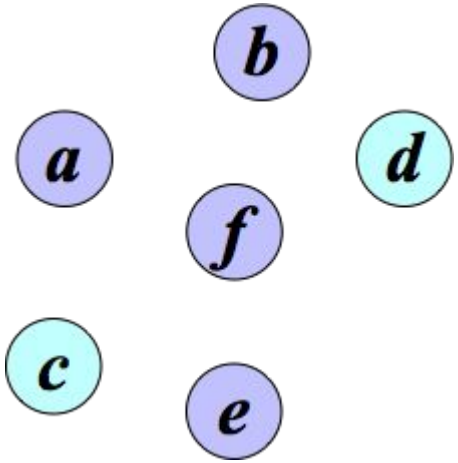


Cycle?

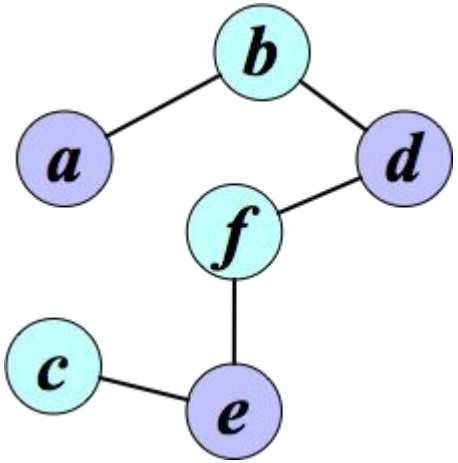


Is a graph bipartite?

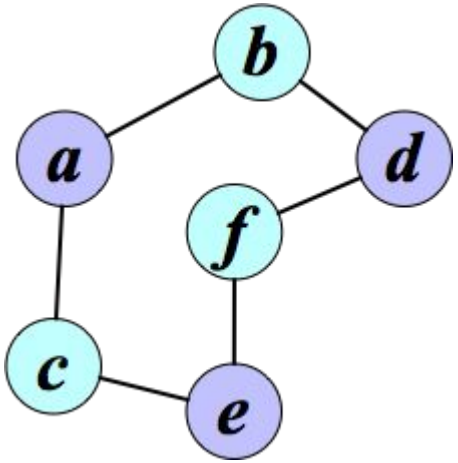
Empty graph?



Path?



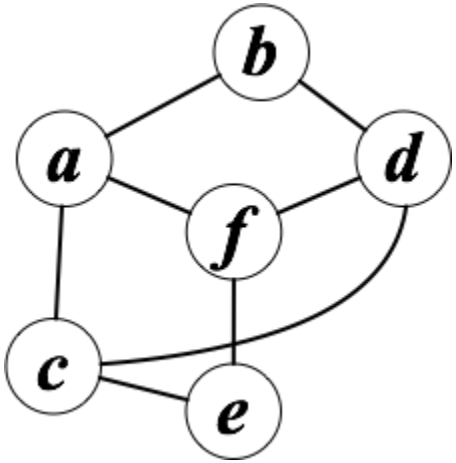
Cycle?

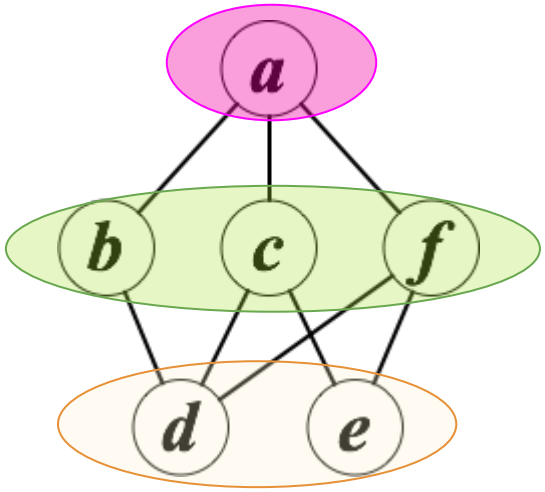


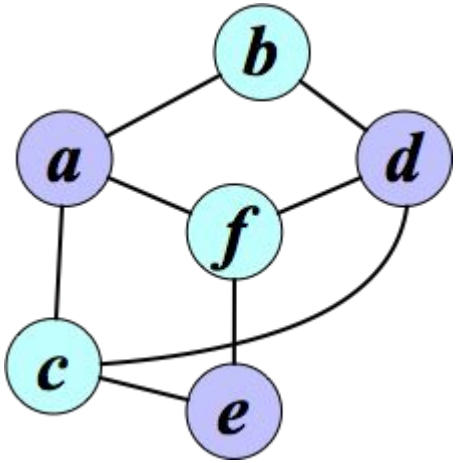
Can we generalize this to an algorithm?

What about…

- BFS from a vertex
- *X*: even layers
- *Y*: odd layers



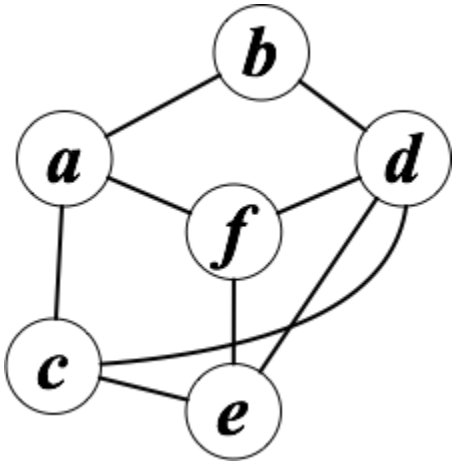


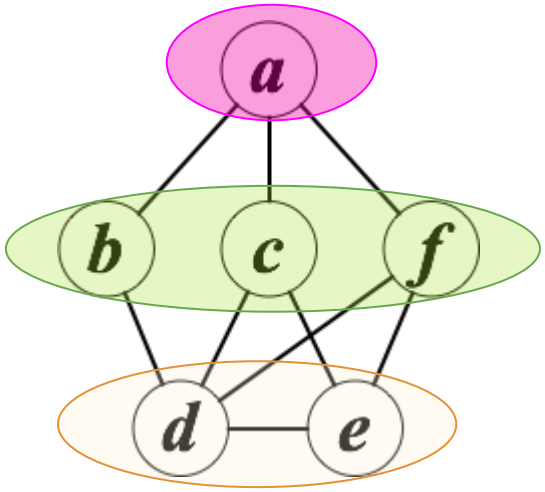


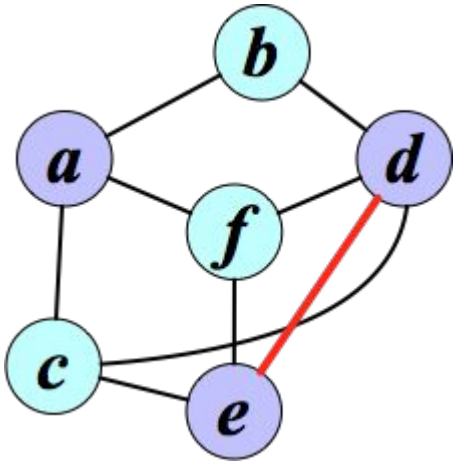
Uh-oh!

What about…

- BFS from a vertex
- *X*: even layers
- *Y*: odd layers

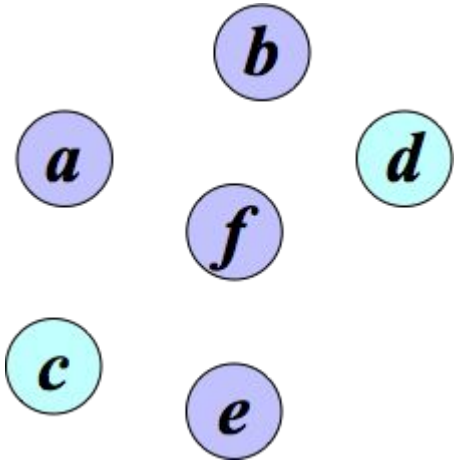




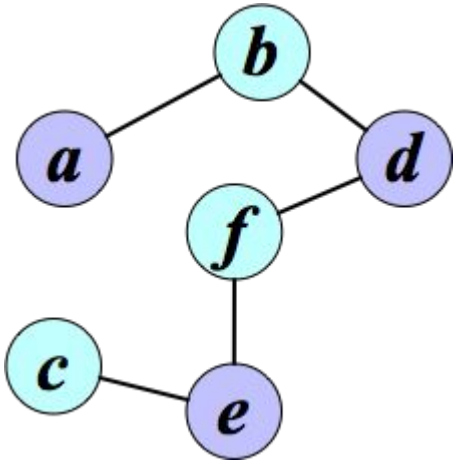


Is a graph bipartite?

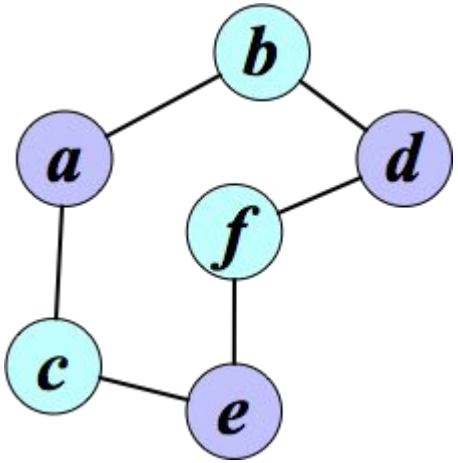
Empty graph?



Path?

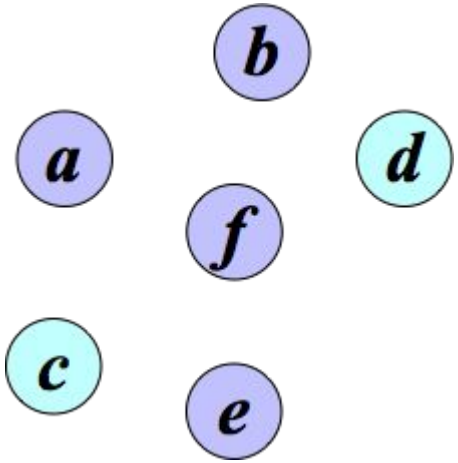


(even) Cycle?

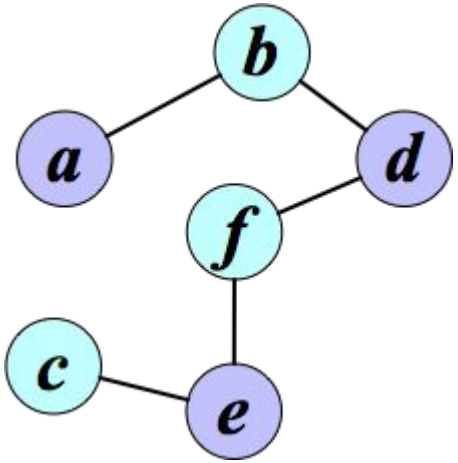


Is a graph bipartite?

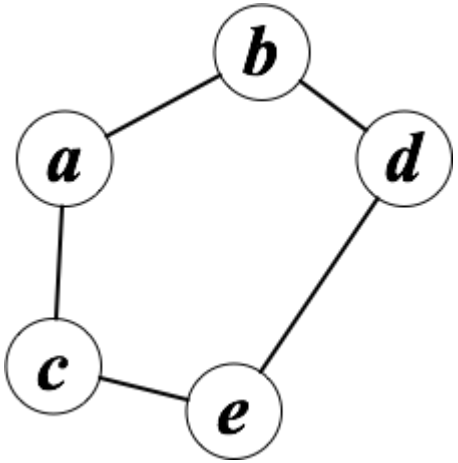
Empty graph?



Path?

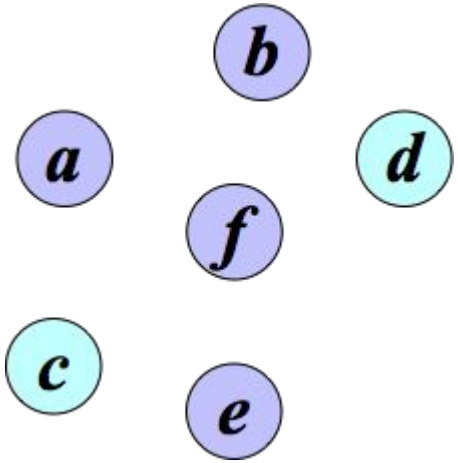


odd
Cycle?

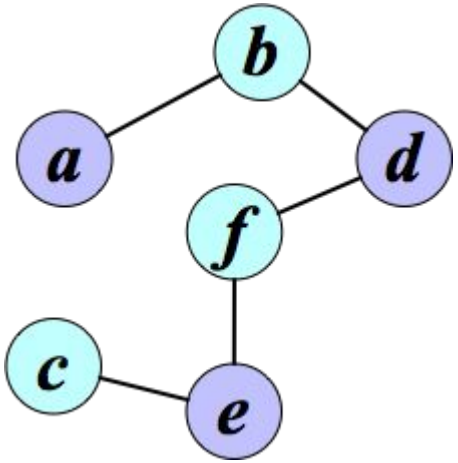


Is a graph bipartite?

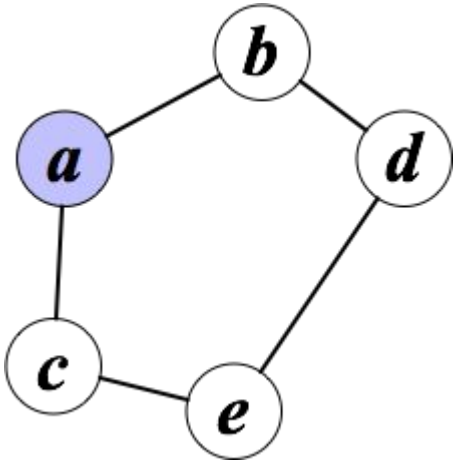
Empty graph?



Path?

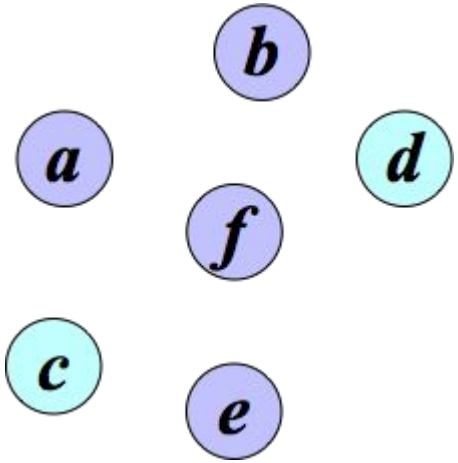


odd Cycle?

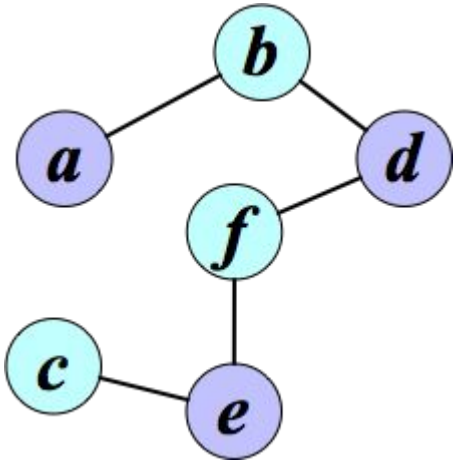


Is a graph bipartite?

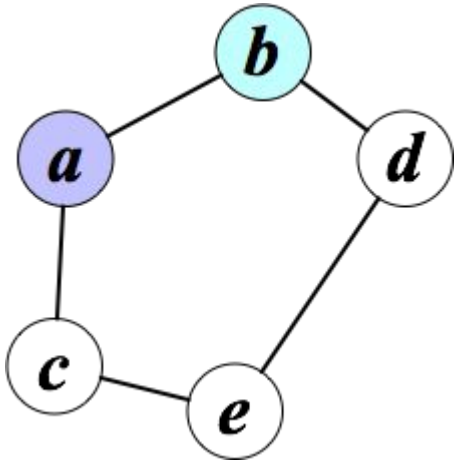
Empty graph?



Path?

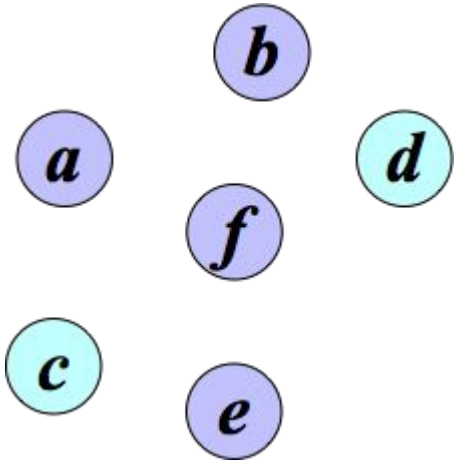


odd Cycle?

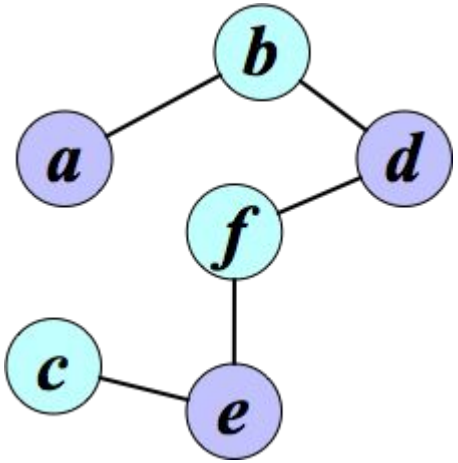


Is a graph bipartite?

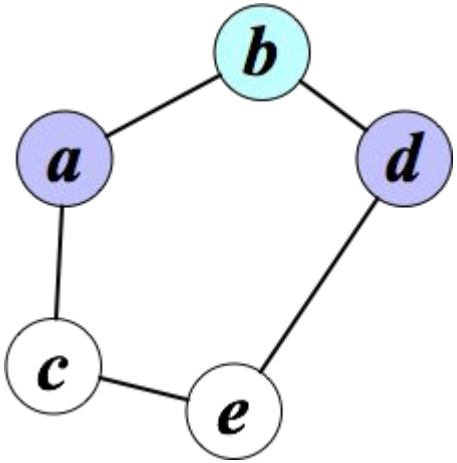
Empty graph?



Path?

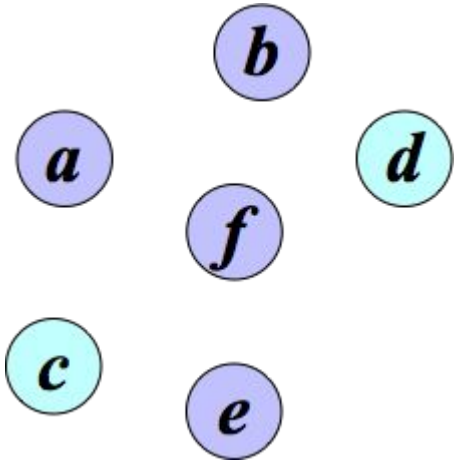


odd Cycle?

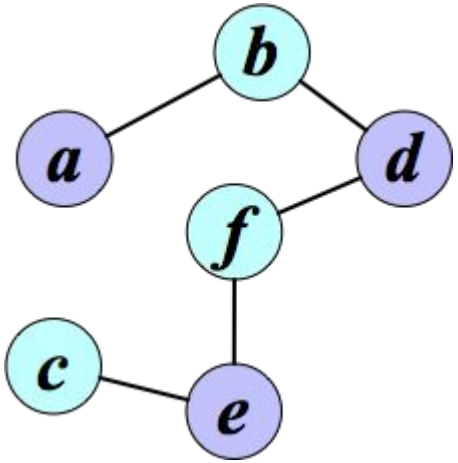


Is a graph bipartite?

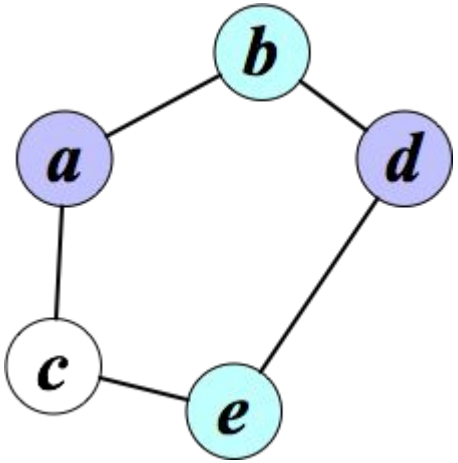
Empty graph?



Path?

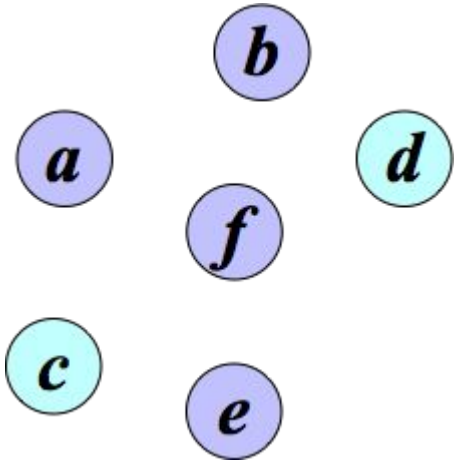


odd
Cycle?

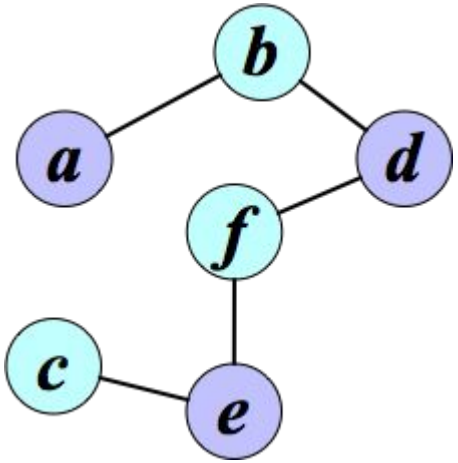


Is a graph bipartite?

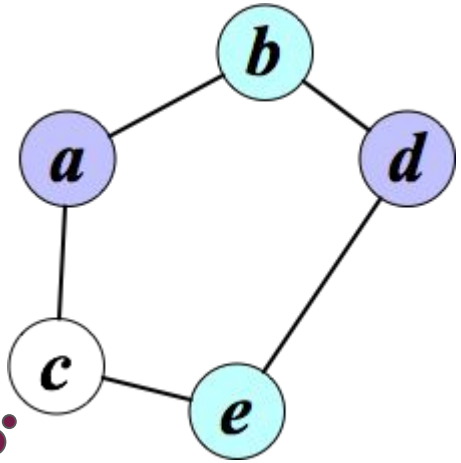
Empty graph?



Path?



odd Cycle?



Bipartite graphs & odd cycles

Claim: Bipartite graphs => no odd cycles

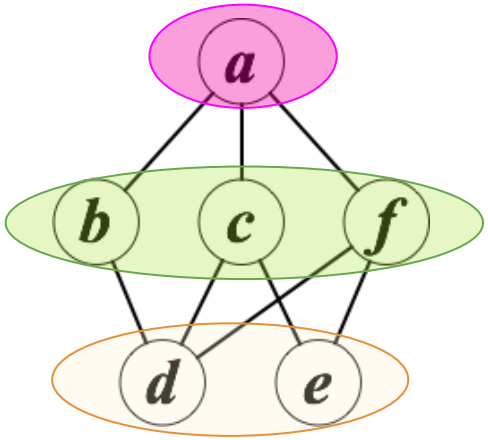
Proof: Via contrapositive: *odd cycle => not bipartite*

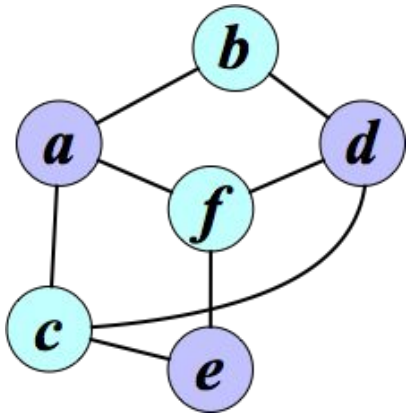
Suppose, for a contradiction, G is bipartite with its vertices partitioned into sets X and Y.
By assumption, G has an odd cycle $P = v_1, v_2, \ldots, v_k$, where k is an even number.
Since there is an edge between each consecutive pair of vertices, the vertices must alternate between X and Y.
WLOG assume $v_1 \in X$; *then* $v_i \in X$ and $v_{i+1} \in Y$ for all odd $i = 1,3,\ldots, k-1$.
In particular, $v_k \in Y$. But, since P is a cycle, $v_1 = v_k$, contradicting the supposition that X and Y are disjoint.

So what happened during our proposed process?

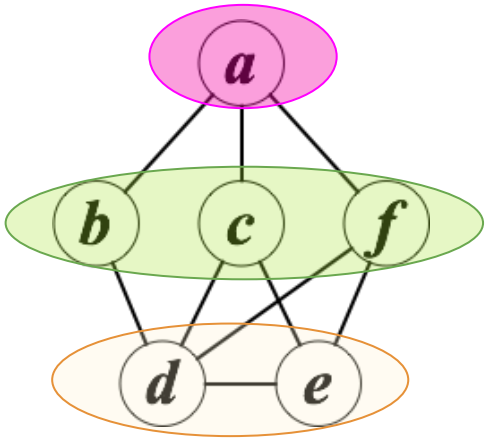
Remember… non-tree BFS edge endpoints are:

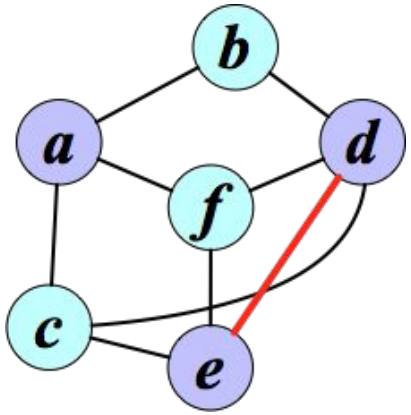
- Same layer, or
- One layer apart





same layer seems problematic?





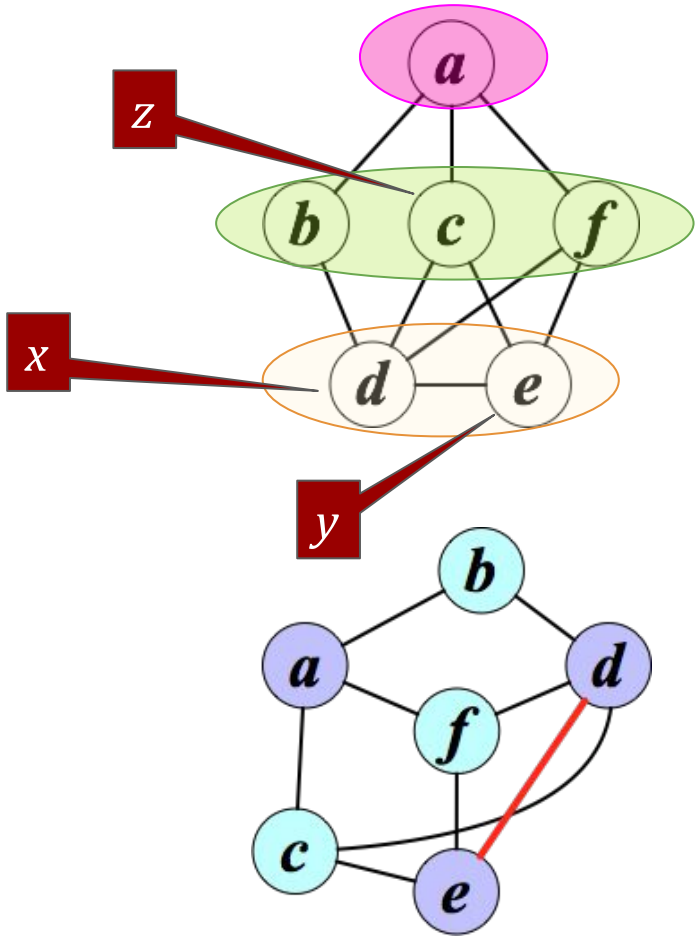
BFS non-tree edge: same layer => odd cycle

Proof:

Let T be the tree output by BFS and let xy be an edge with both x and y in layer L_i.

Consider z, the least common ancestor of x and y, in layer L_j.

The cycle z-xy-z is of length
$(i\text{-}j) + 1 + (i\text{-}j) = 2*(i\text{-}j) + 1$,
so the graph contains an odd cycle.
Therefore, it is not bipartite.



We have an algorithm!

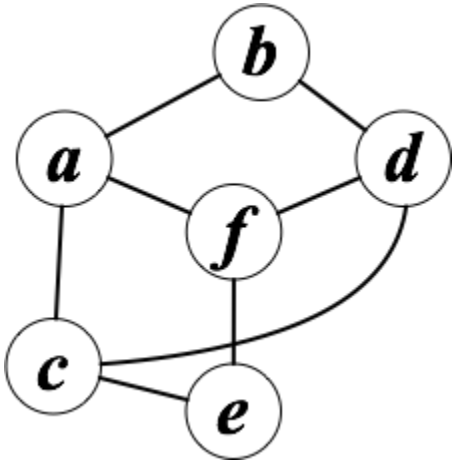
isBipartite:

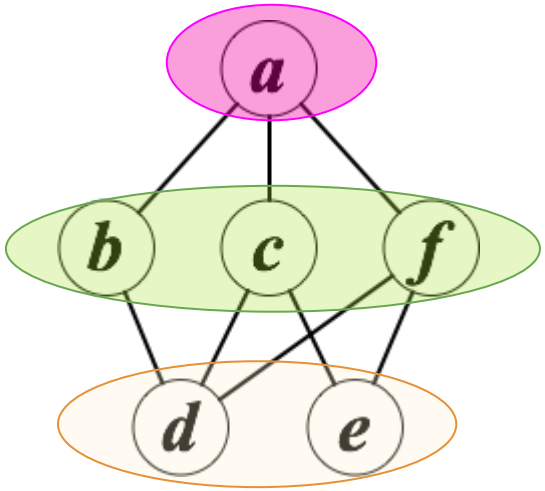
construct BFS tree from any vertex

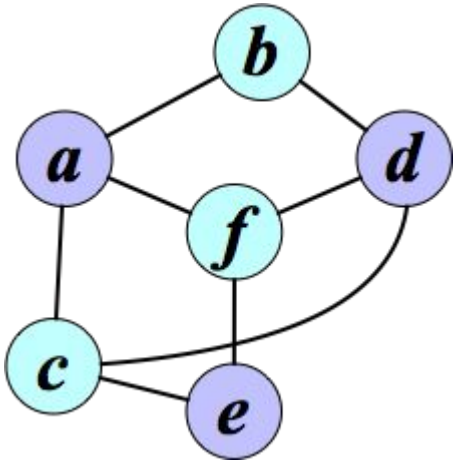
for each layer

if exists non-tree edge, return FALSE

return TRUE with (*X*: even layers, *Y*: odd layers)







Correctness

Claim: *G* is bipartite if and only if **isBipartite** returns TRUE

Proof:

(=>) Show: *G* is bipartite => **isBipartite** returns TRUE
Via contrapositive: **isBipartite** returns FALSE => *G* is not bipartite.
isBipartite returns FALSE => non-tree edge in same layer of BFS tree
=> odd cycle => *G* is not bipartite

(<=) Show: **isBipartite** returns TRUE => *G* is bipartite
Tree edges alternate *X* and *Y*.
BFS non-tree edges between adjacent layers or in same layer.
    Adjacent layers respect bipartite
    no edges in same layer <=> **isBipartite** returns TRUE
G is bipartite.

Running time

isBipartite:

construct BFS tree from any vertex

for each layer

if exists non-tree edge, return FALSE

return TRUE with (X: even layers, Y: odd layers)

- BFS: $O(m + n)$
- Good data structure → non-tree edge check if same layer: $O(m)$
  - array to label vertex with layer #
- Total: $O(m + n)$