Algorithms on Graphs

Umang Bhaskar & Juhi Chaudhary

Day 1, session 2: Proofs on graphs

STCS Vigyan Vidushi 2024

Exercises: Day 1

Problem 1:

Give a polynomial time algorithm to test if a graph is bipartite.

Exercises: Day 1

Problem 1:

Give a polynomial time algorithm to test if a graph is bipartite.

- show that G is bipartite iff G has no odd cycles

Exercises: Day 1

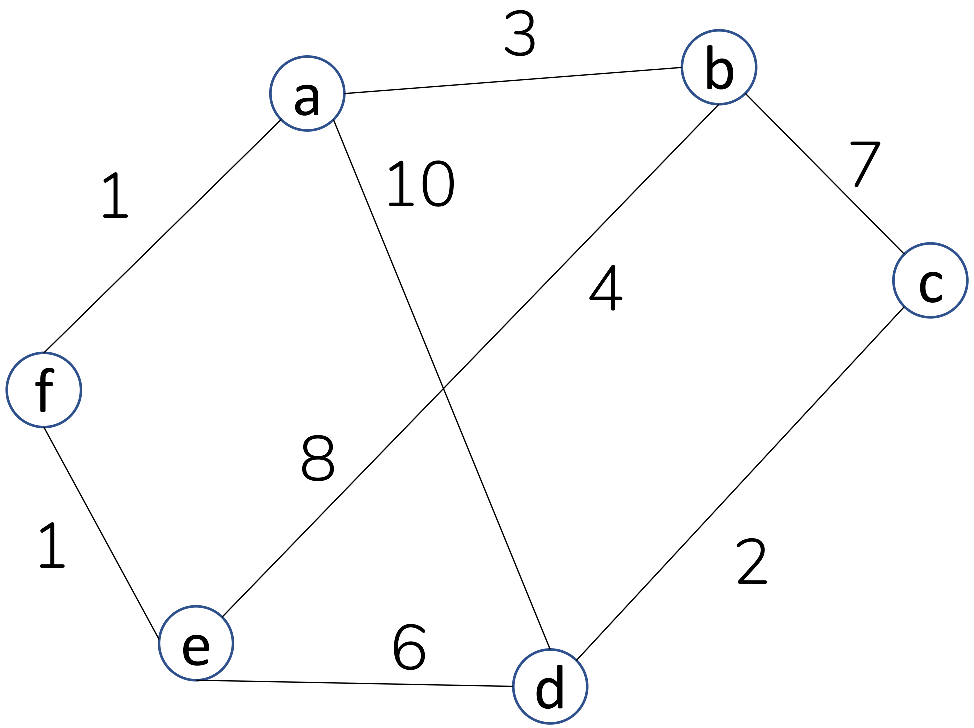
Problem 1:

Give a polynomial time algorithm to test if a graph is bipartite.

- show that G is bipartite iff G has no odd cycles
- use bfs to find odd cycles

Problem 2:

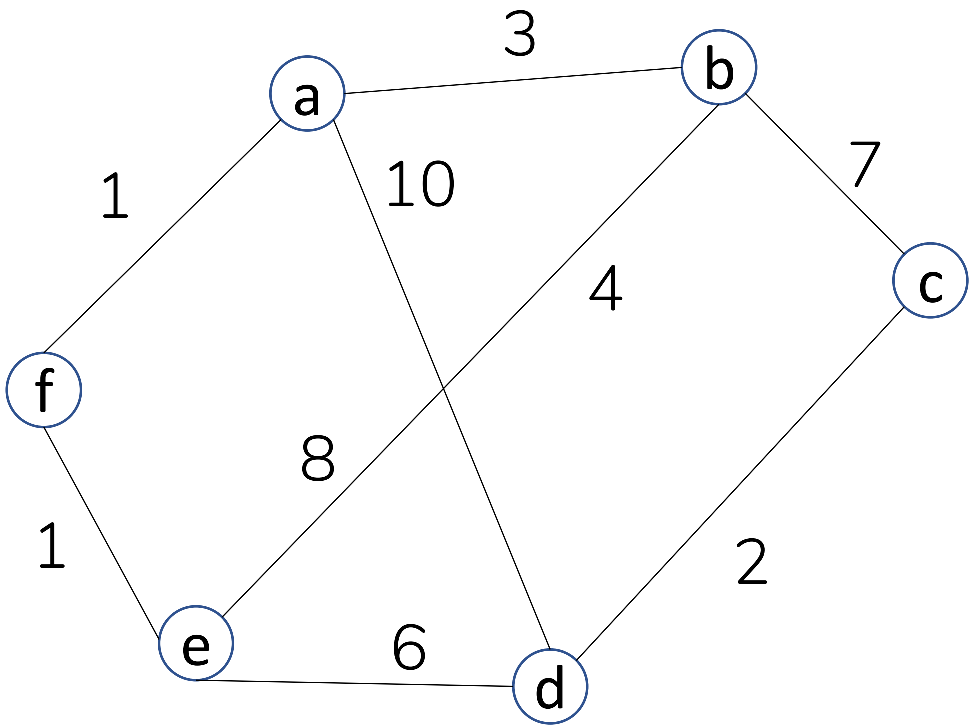
Show that every MST must include both edges with weight 1.



Problem 2:

Show that every MST must include both edges with weight 1.

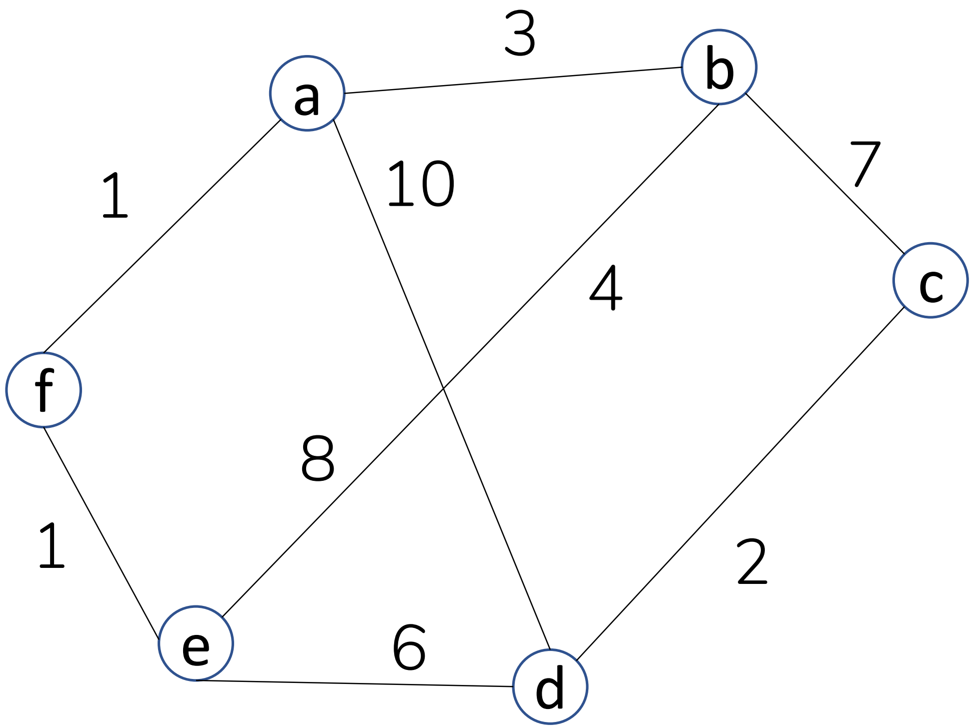
- how do you characterize a spanning tree?



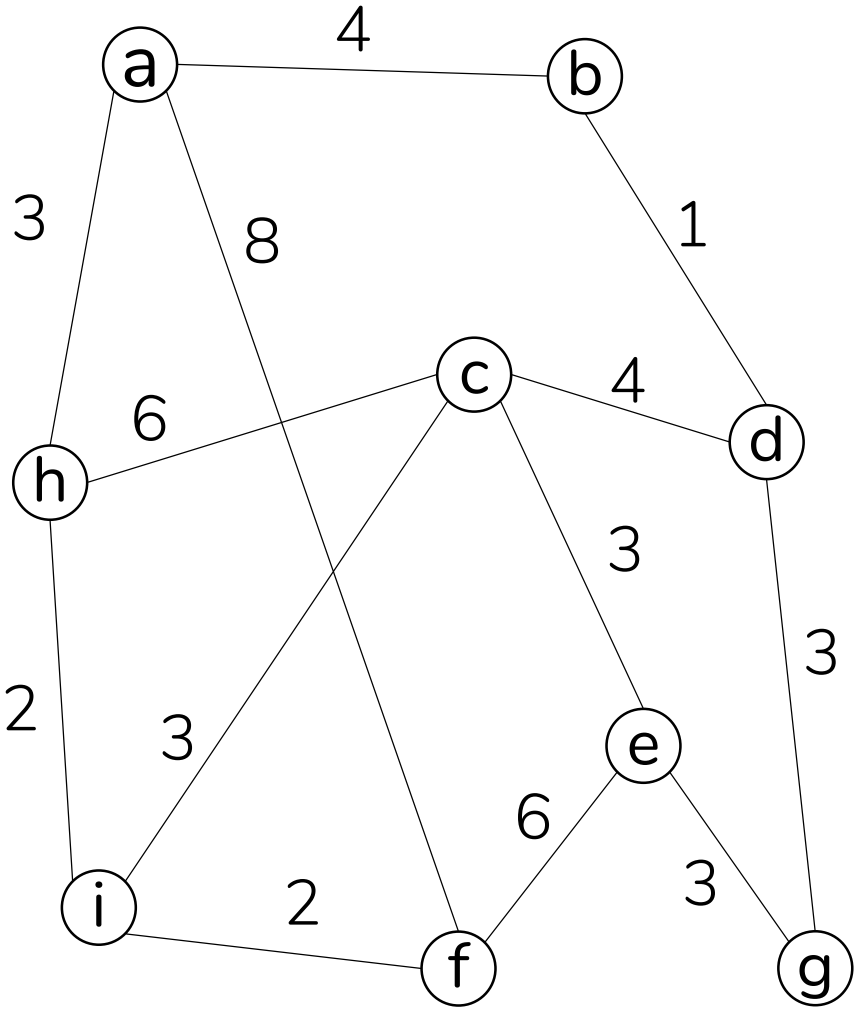
Problem 2:

Show that every MST must include both edges with weight 1.

- how do you characterize a spanning tree?
- assume T is an MST that doesn't contain both edges. What happens when you add the missing weight 1 edge?



Problem 3: Find all shortest-paths from vtx a



Dijkstra ($G = (V, E)$, a)

---

dist(a) = 0, dist(v) = ∞ for all others

S = {a} \\ set of shortest-distance vertices

update distance for vtxs adjacent to a

while S ≠ V

add vtx v with smallest dist to S

update distance for vtxs w adjacent to v

add directed edge (v,w)
(and remove other edges into w)

Problem 4:

Prove that Dijkstra’s algorithm terminates correctly.

Problem 4:

Prove that Dijkstra’s algorithm terminates correctly.

- prove that it terminates.

Problem 4:

Prove that Dijkstra’s algorithm terminates correctly.

- prove that it terminates.
- prove that when a vertex is added to S, it’s distance is shortest-path distance.