COMP 3711

Tutorial 7

Question 1

Let G=(V,E) be an undirected graph where
V is the set of vertices and E is the set of edges.

Assume that there are no self-loops or duplicated edges.

Answer all questions below as a function of $|V|$, the number of vertices.

a) What is the maximum number of edges in G?

b) What is the maximum number of edges in G if two vertices have degree 0?

c) What is the maximum number of edges that an acyclic graph G can have?

d) What is the minimum number of edges in G ,
if G is a connected graph and contains at least one cycle?

e) What is the minimum possible degree that
a vertex in a connected graph G can have?

f) What is the maximum length of any simple path in G?

Solution 1

a) What is the maximum number of edges in any graph G?

$$\binom{|V|}{2} = \frac{|V|(|V|-1)}{2}$$

This is the number of pairs that can be made out of $|V|$ vertices (each pair can be an edge).

b) What is the maximum number of edges in G if two vertices have degree 0?

$$\binom{|V|-2}{2} = \frac{(|V|-2)(|V|-3)}{2}$$

This is the number of pairs among the $|V|-2$ points that can have edges.

Note that answer to both (a) and (b) are $\Theta(V^2)$
$\Rightarrow$ asymptotic answer to both questions is $O(V^2)$.
Since we sometimes use *n* to denote *V* this is sometimes written as $O(n^2)$

Solution 1 (cont'd)

c) What is the maximum number of edges in an acyclic graph G?

$$|V| - 1$$

This occurs when G is connected. We learned in class that a connected acyclic graph (a tree) has $|V| - 1$ edges.
If the acyclic graph is not connected it has even fewer edges

d) What is the minimum number of edges in G if G is a connected graph and contains at least one cycle?

$$|V|$$

A connected graph has $|V| - 1$ edges and this occurs if and only if it is acyclic (from class). Adding one edge forms a cycle.

e) What is the minimum possible degree of a vertex in a connected graph G?

For $|V| < 2$, minimum degree is 0.

For $|V| \geq 2$, minimum degree is 1, otherwise the graph will not be connected.

Solution 1 (cont'd)

f) What is the maximum length of any simple path in G?

$$|V| - 1$$

The longest simple path will traverse all vertices in G exactly once, so the length of the path is $|V| - 1$.

Note: While the above might seem like "academic" questions, they often arise in the analysis of algorithms. More specifically the running times of graph algorithms are often functions of both $|V|$ and $|E|$.

If you are asked to design an algorithm with V vertices and specific constraints on the graph, knowing how many edges the graph has will often lead a better analysis of the algorithm's running time.

Question 2

Let $G = (V, E)$ be a connected undirected graph. Prove that

$$\log(|E|) = \Theta(\log |V|).$$

Note: we implicitly use this fact in many of our analyses in class.

Solution 2

First note that $|E| \le |V|^2$. Thus

$$\log|E| \le \log|V|^2 = 2\log|V|$$

=> $\log|E| = O(\log|V|)$.

Next, because the graph is connected, $|V| - 1 \le |E|$

=> If $|V| \ge 4$ $\quad \frac{|V|}{2} \le |V| - 1 \le |\mathrm{E}|$.

=> $\log|\mathrm{V}| - 1 = \log|V|/2 \le \log|E|$

=> $\log|E| = \Omega(\log|V|)$

Combining the two directions gives

$$\log|E| = \Theta(\log|V|).$$

Question 3

The adjacency list representation of a graph G, which has 7 vertices and 10 edges, is:

$a: \rightarrow d, e, b, g$

$b: \rightarrow e, c, a$

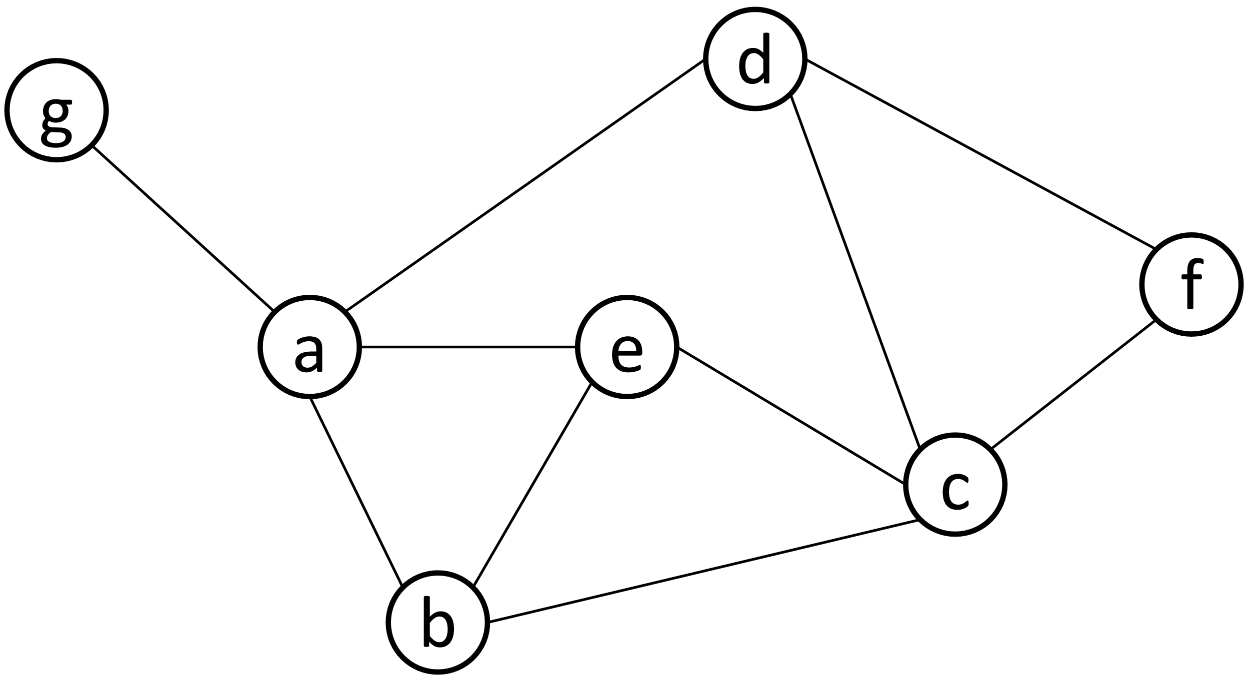
$c: \rightarrow f, e, b, d$

$d: \rightarrow c, a, f$

$e: \rightarrow a, c, b$

$f: \rightarrow d, c$

$g: \rightarrow a$



Question 3 (cont'd)

(a) Show the breadth-first search tree that is built by running BFS on graph G with the given adjacency list, using vertex a as the source.

(b) Indicate by dashed lines the edges in G that are NOT in the BFS tree in part (a).

(c) Show the depth-first search tree that is built by running DFS on graph G with the given adjacency list, using vertex a as the source.

(d) Indicate by dashed lines the edges in G that are NOT in the DFS tree in part (c).

Solution 3: (a) and (b)

BFS

$a\!:\rightarrow d, e, b, g$

$c\!:\rightarrow f, e, b, d$

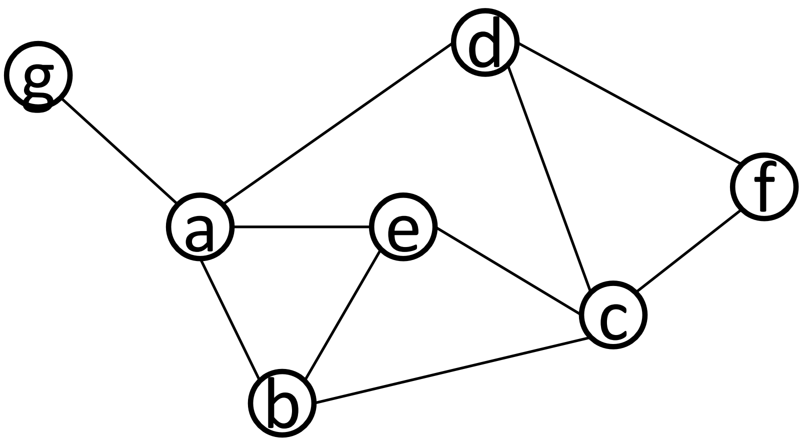
$e\!:\rightarrow a, c, b$

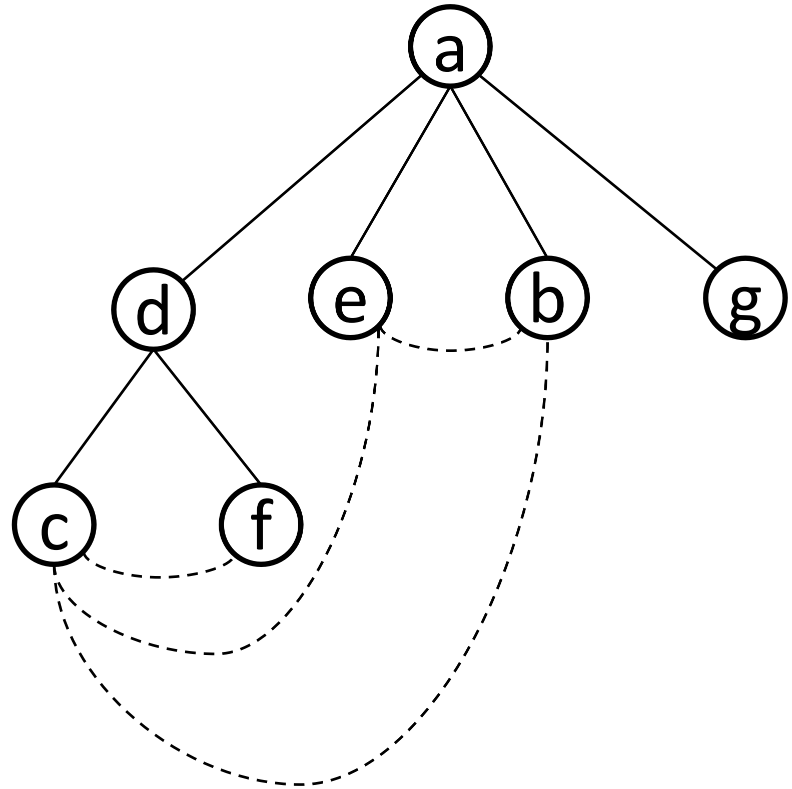
$g\!:\rightarrow a$

$b\!:\rightarrow e, c, a$

$d\!:\rightarrow c, a, f$

$f\!:\rightarrow d, c$





Solution 3: (c) and (d)

DFS

$a\!:\rightarrow d, e, b, g$

$c\!:\rightarrow f, e, b, d$

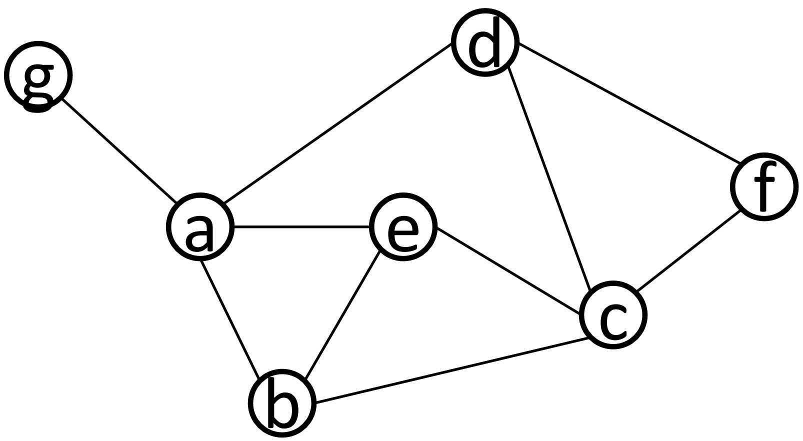
$e\!:\rightarrow a, c, b$

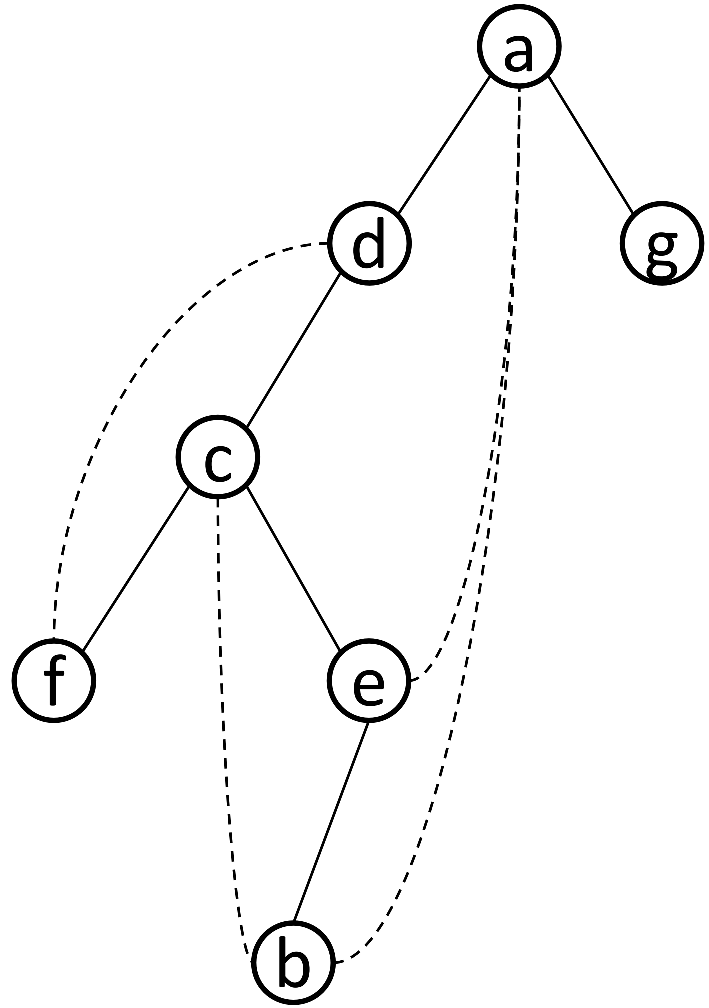
$g\!:\rightarrow a$

$b\!:\rightarrow e, c, a$

$d\!:\rightarrow c, a, f$

$f\!:\rightarrow d, c$



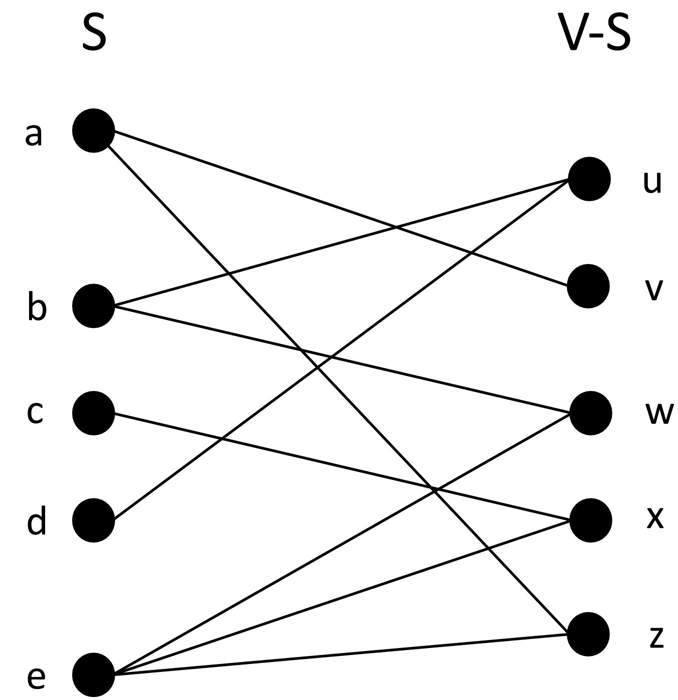


Question 4

An (undirected) graph G=(V,E) is ***bipartite*** if there exists some $S \subset V$ such that, for every edge $\{u, v\} \in E$, either

(i) $u \in S, v \in V - S$ or

(ii) $v \in S, u \in V - S$.



Let G=(V,E) be a connected graph.

Design an O(|V|+|E|) algorithm that checks whether G is bipartite.

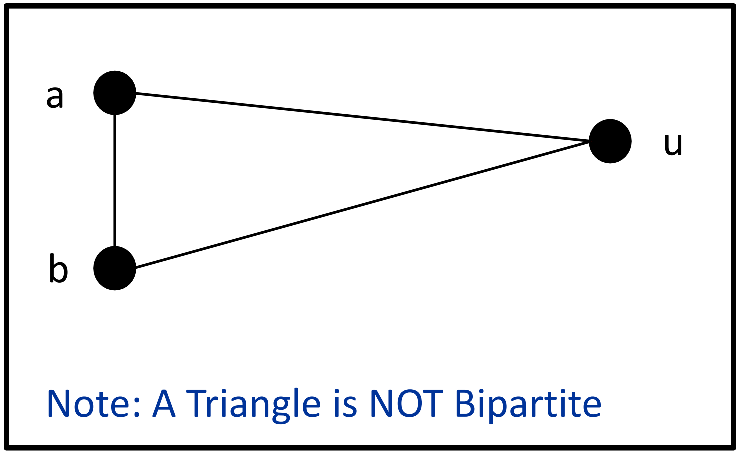
Hint: Run BFS.

Question 4

An (undirected) graph G=(V,E) is ***bipartite*** if there exists some $S \subset V$ such that, for every edge $\{u, v\} \in E$, either

(i) $u \in S, v \in V - S$ or

(ii) $v \in S, u \in V - S$.



Let G=(V,E) be a connected graph.

Design an O(|V|+|E|) algorithm that checks whether G is bipartite.

Hint: Run BFS.

Solution 4

Let G=(V,E) be a connected graph.
Design an O(|V|+|E|) algorithm that checks whether G is bipartite.

Run BFS from any vertex.
Recall that d[v] will store the shortest distance from root v.

Set S to be the set of all vertices with d[v] even.

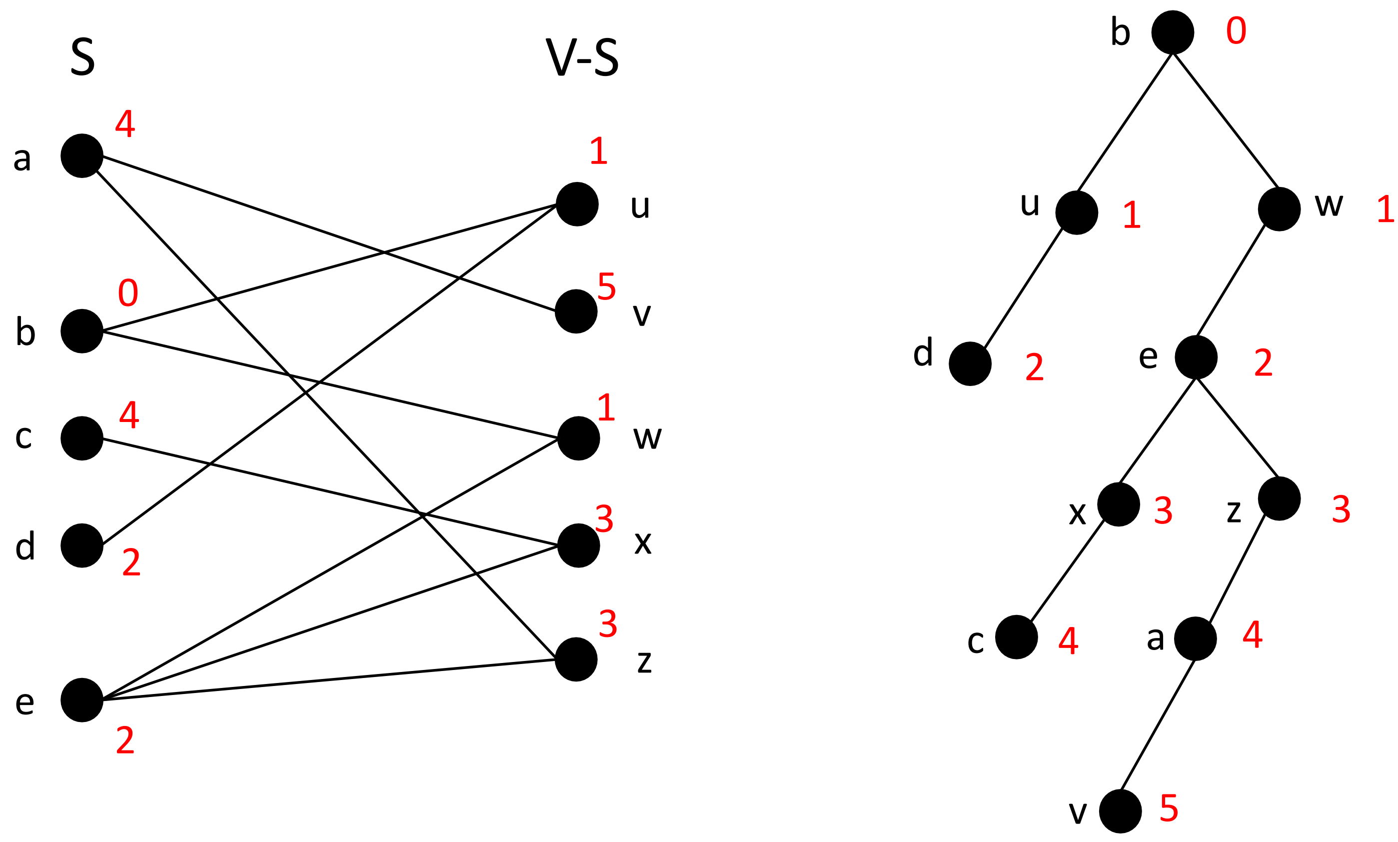
G will be bipartite
if and only if
all edges (u, v) satisfy that the parity of d[v] and d[u] are different,

i.e., d[v] is odd and d[u] is even or vice versa.

If the observation is correct the algorithm is correct!

Solution 4 (cont'd)

A BFS search tree from b:



On the next slides we will prove that G is bipartite

if and only if

all edges (u, v) satisfy that the parity of d[v] and d[u] are different,

Solution 4 (cont'd)

(a) G will be bipartite
if and only if
(b) all edges (u, v) satisfy that the parity of d[v] and d[u] are different,

i.e., d[v] is odd and d[u] is even or vice versa.

(a) <= (b) If (b) is true, it is easy to see that G is, by definition bipartite. Just set S to be the set of all even vertices.

(a) => (b) Now suppose that the graph is bipartite. Let $S, V - S$ be the bipartite split and assume without loss of generality that $w \in S$.

Then **by definition of bipartite**,
the length of *every* path from w to other nodes in S is even &
the length of *every* path from w to nodes in $V - S$ is odd.

In particular the lengths of the *shortest* paths to nodes in S are even and the lengths of the *shortest* paths to nodes in $V - S$ are odd.
⇒ the parity of the endpoints of all edges in G must be different.

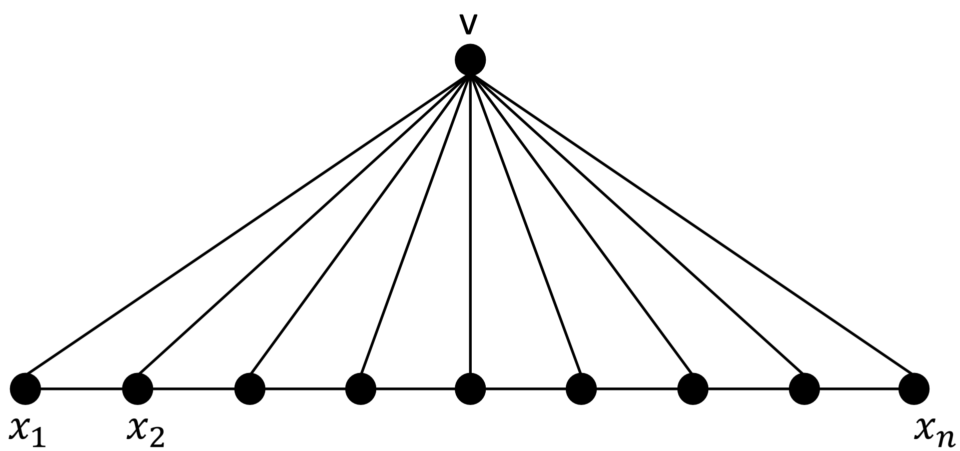
Question 5

In the Fan Graph F_n, node v is connected to all the nodes and the other connections are given by the adjacency lists below.

$$v\colon \rightarrow x_1, x_2, \ldots, x_n$$

$$x_1\colon \rightarrow v, x_2 \qquad\qquad x_n\colon \rightarrow v, x_{n-1}$$

$$\forall i \neq 1, n, \qquad x_i\colon \rightarrow v, x_{i-1}, x_{i+1}$$

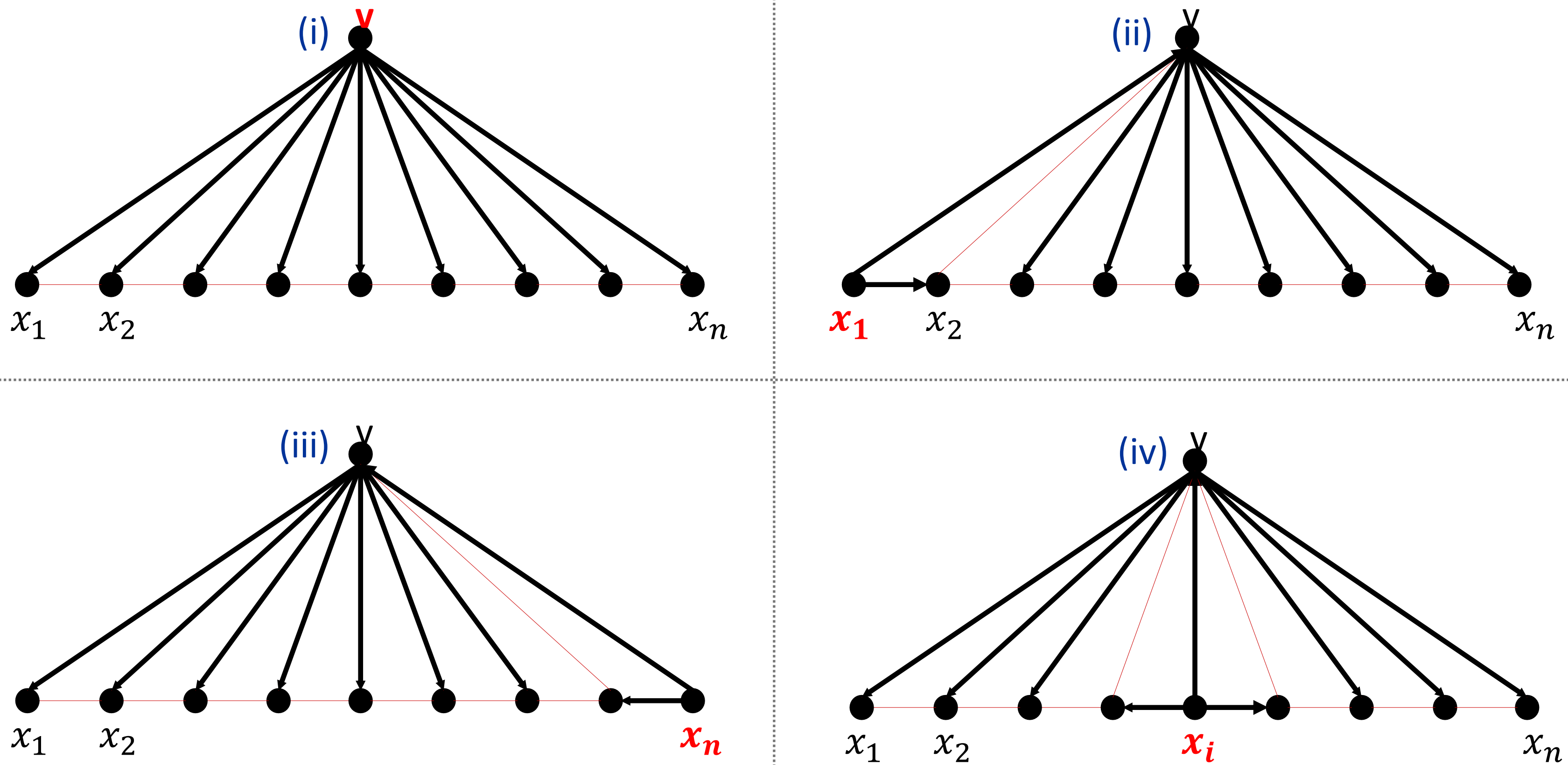


(a) Describe the tree that is output when BFS is run on F_n starting from vertex: (i) v; (ii) x_1; (iii) x_n; (iv) Other x_i.

(b) Describe the tree that is output when DFS is run on F_n starting from vertex: (i) v; (ii) x_1; (iii) x_n; (iv) Other x_i.

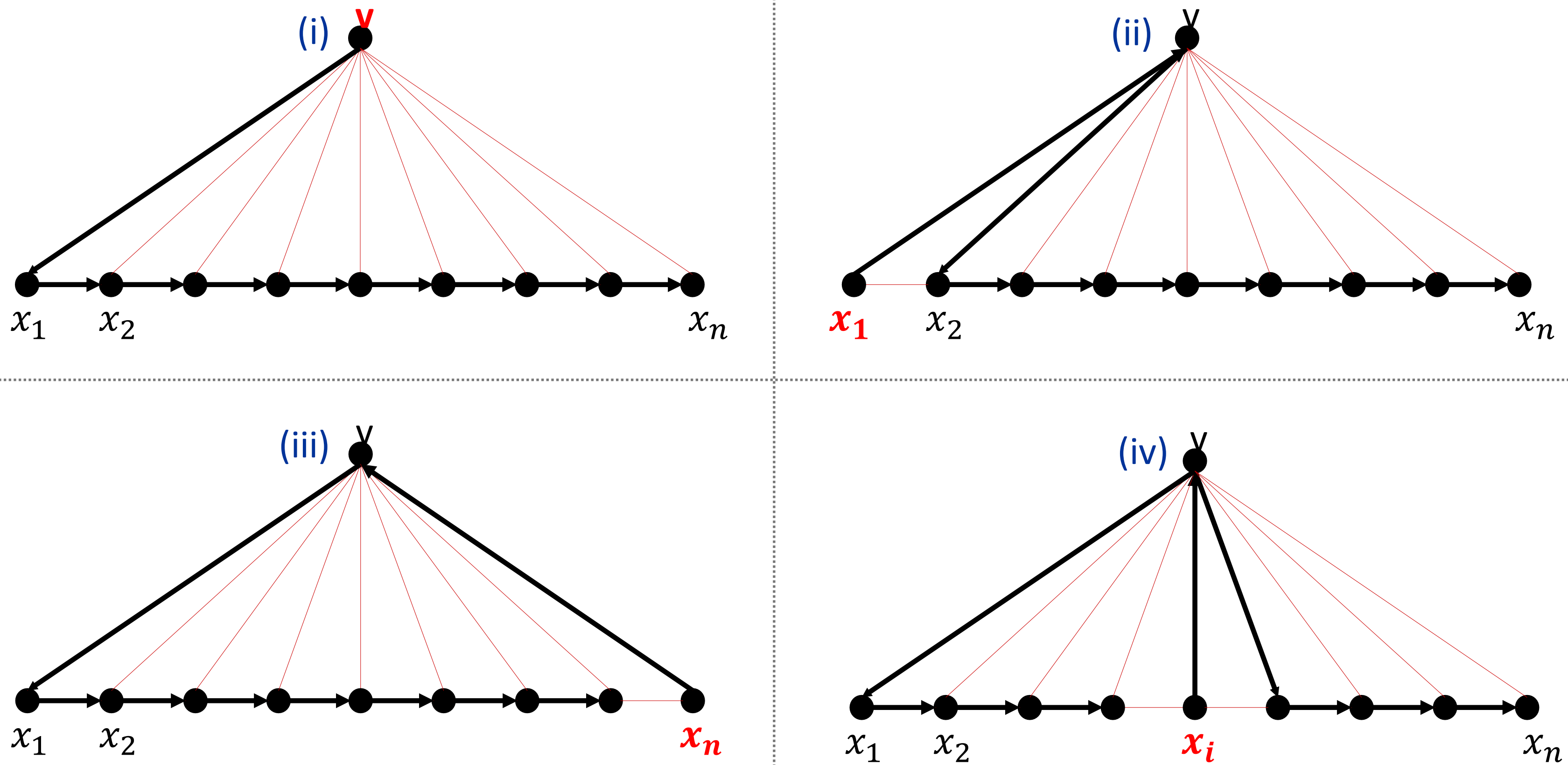
Solution 4: BFS

Describe the tree that is output when BFS is run on F_n starting from vertex: (i) v; (ii) x_1; (iii) x_n; (iv) Other x_i.



Solution 5: DFS

Describe the tree that is output when DFS is run on F_n starting from vertex: (i) v; (ii) x_1; (iii) x_n; (iv) Other x_i.



More

The purpose of this problem was to illustrate that the output of BFS and DFS doesn’t only depend on the graph and the starting point.

It also depends strongly on the graph representation, i.e. the orderings of each adjacency list.

Different orderings of the same list can lead to radically different BFS and DFS trees!