12.1 Exercise 4:

If $G=(V,E)$ is a forest with $|V|=v$, $|E|=e$ and k connected components, find a relationship among v, e, k.

Solution:

Let $T_1, \dots, T_k$ be the different components and $v_i = |V_i|$, $e_i = |E_i|$ the number of vertices and edges of T_i.

We know that for each i $v_i - e_i = 1$.

Also $v = \sum_{i=1}^{k} v_i$, $e = \sum_{i=1}^{k} e_i$,

so $\boxed{v - e = \sum_{i=1}^{k} v_i - e_i = k}$.

Exercise 6: a) Verify that all trees are planar.

b) Derive Theorem 3 from a).

Solution:

a) Informally, the idea is that, since the graph has no cycle, we can just place the vertices one after the other in such a way that the edges do not intersect (for example choosing a root and starting from there).

Formally, it is enough to prove that a tree has no embedded K_5 or $K_{3,3}$. However both graphs have a cycle and a tree does not, so we cannot embed them in a tree.

(Kuratowski theorem)

b) Recall that the Euler formula states

$$|V| - |E| + |F| = 1 .$$

Since there is no cycle in a tree, we also have no face, and thus $|F| = 0$.

This gives us Theorem 3: $|V| - |E| = 1$.

Exercise 12:

Let $G=(V,E)$ be a loop-free connected undirected graph where $V=\{v_1,v_2,\dots,v_n\}$ $n\geq 2$ $\deg(v_1)=1$ and $\deg(v_i)\geq 2$ for $2\leq i\leq n$.
Prove that G must have a cycle.

We use the formula for the number of edges,

$$2\cdot|E| = \sum_{v\in V}\deg(v) = 1+\sum_{i=2}^{n}\deg(v_i)$$

$$\geq 1+\sum_{i=2}^{n} 2 = 1+2(n-1) = 2n-1$$

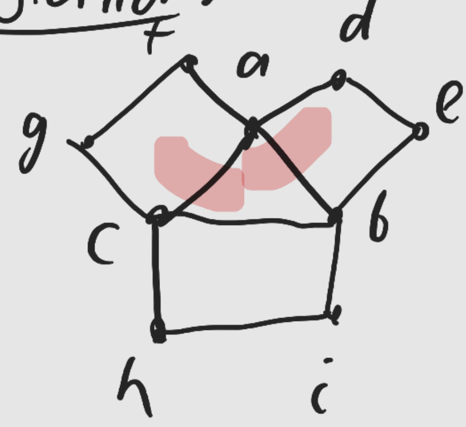
Therefore $|E|\geq \frac{2n-1}{2}$, so $|E|\geq n$.

However this means that $|V|-|E|\leq n-n=0$, and so the graph cannot be a tree, since it would require $|V|-|E|=1$.

Exercise 16:

For the following, determine how many non-identical (but maybe isomorphic) spanning trees exist:

<u>Solution:</u>

(1) 

We count $|V| = 9$ and $|E| = 12$, so we will need to delete 4 edges to get a spanning tree. Also in each of the square we can delete at most one outer edge otherwise the graph gets disconnected.

We distinguish according to how many edges of the triangle abc get deleted:

- <u>3 edges</u>: Then we need to delete one additional edge, and there are 9 possibilities
- <u>2 edges</u>: There 3 ways of choosing those two, then we need to delete an edge in the square of the third edge and another edge in the remaining cycle.

There are respectively 3 and 6 possibilities, so in total this makes $3 \cdot 3 \cdot 6 = 54$ possibilities.

- <u>1 edge</u>: There are 3 ways of choosing these edge, then we need to remove one edge in each of the squares, so there are $3 \cdot 3^3 = 81$ possibilities

- <u>0 edges</u>: We don't get a tree.

In total, there are $\boxed{9 + 54 + 81 = 144 \text{ spanning trees}}$

(2) By the same argument, we get

$8 + 4 \cdot 2 \cdot 6 + 6 \cdot 2^2 \cdot 4 + 4 \cdot 2^3 = \boxed{184 \text{ spanning trees}}$

(3) By the same argument, we get

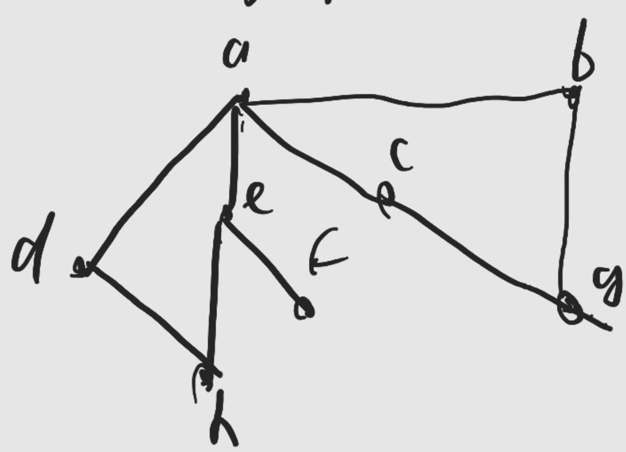
$16 + 4 \cdot 4 \cdot 12 + 6 \cdot 4^2 \cdot 8 + 4 \cdot 4^4 = \boxed{2000 \text{ spanning trees}}$

12.2 Exercise 7:

a) Find the depth-first spanning tree in the following graph if the order of the vertices is
i) a, b, c, d, e, f, g, h
ii) h, g, f, e, d, c, b, a
iii) a, b, c, d, h, g, f, e.

Solution:

The original graph is



i)

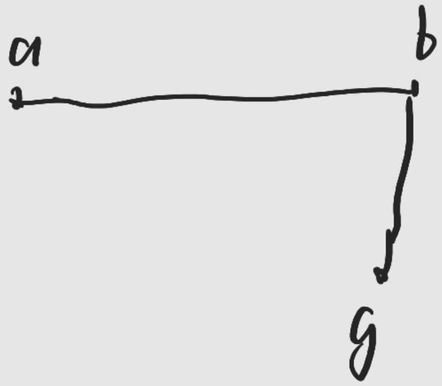
Step 1: $v = a$

$T =$ 

Step 2.0: The first vertex which neighbours a and has not yet been visited is b, so we attach $\{a,b\}$ to T and we assign $v=b$ and return to Step 2:

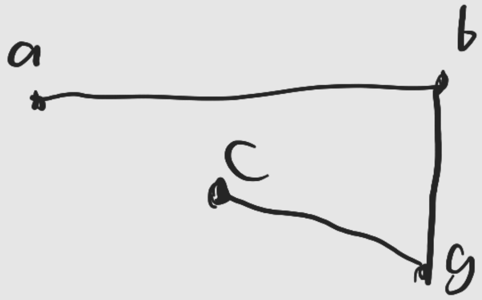
$v=b$ $T=$ 

<u>Step 2.1:</u> The next vertex visited is g..

$v=g$
$T=$ 

<u>Step 2.2</u>: The next vertex is c.

$v=c$

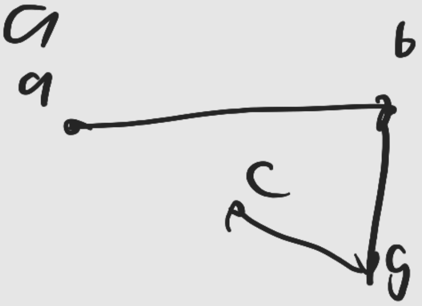
$T=$ 

<u>Step 2.3</u>: There is no neighbor of c that hasn't been visited yet, so we go to Step 4 since $v \neq v_1$.

<u>Step 4</u>: We backtrack to the parent of c in T, i.e. the vertex g, and actually we

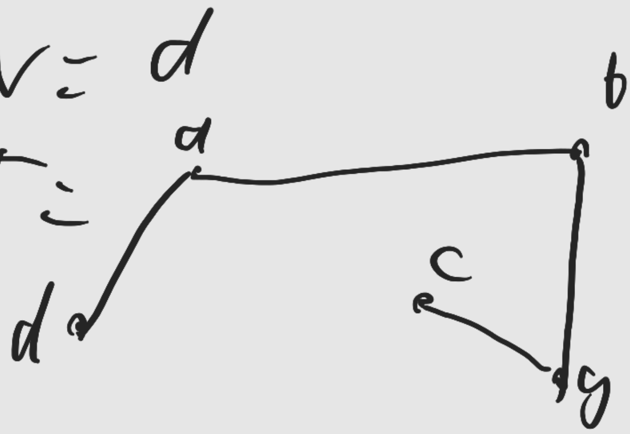
need to backtrack all the way back to a.

$V = a$

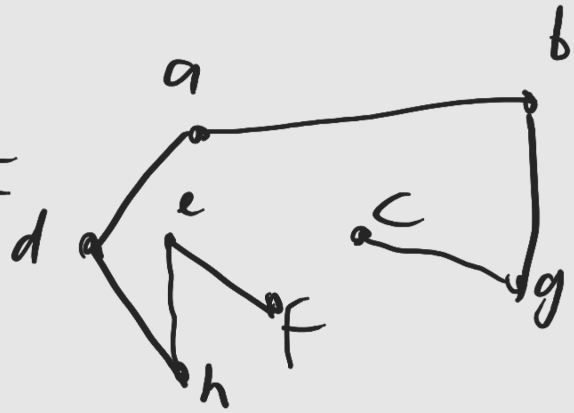
$T =$ 

<u>Step 2.4</u>: The next vertex is d

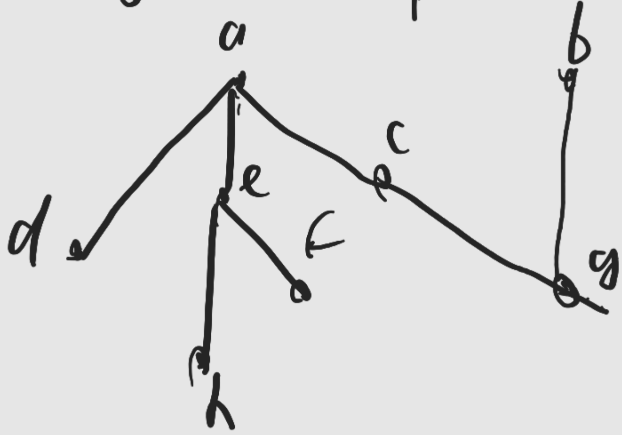
$V = d$

$T =$ 

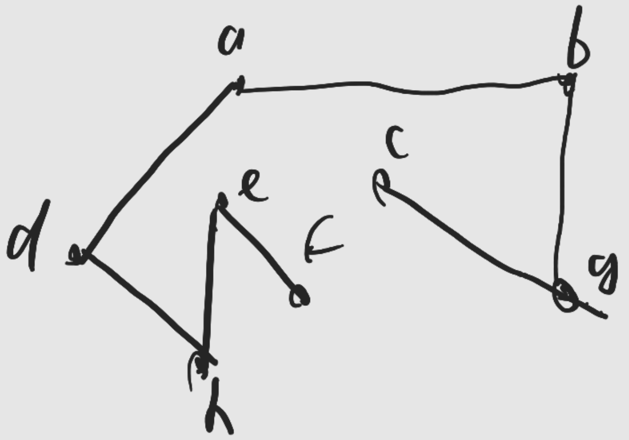
We continue the algorithm until we reach

$T =$ 

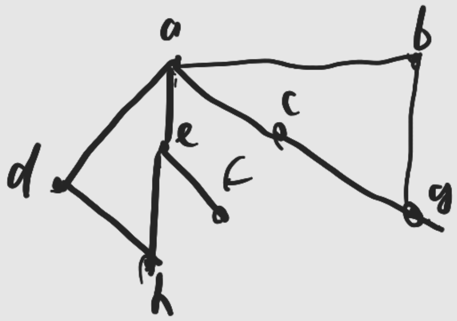
(ii) We get the spanning tree



iii) We gett the spanning tree



Exercise 8: Find the breadth-first spanning tree.



Solution:

i) Step 1: $Q = [a]$ we visit a

$T = \overset{a}{\bullet}$

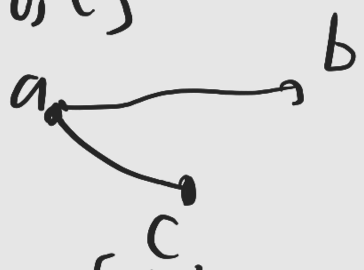
visited $= \{a\}$

Step 2: We delete a from Q and consider, in order, all vertices adjacent to a. The first one is b, it has not been visited yet, we insert b in Q: $Q = [b]$,

$T = \overset{a}{\bullet} \longrightarrow \overset{b}{}$ and we visit b, visited $\{a, b\}$

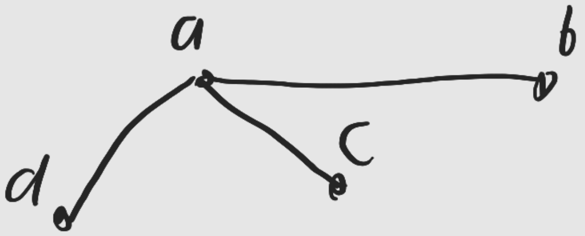
the next vertex neighboring a is c, it has not been visited, so we add c to Q:

$Q = [b, c]$

$T =$ 

visited = $\{a, b, c\}$

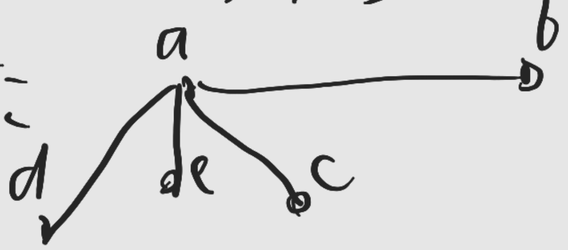
The next vertex d has not been visited

$Q = [b, c, d]$ $T =$ 

visited = $\{a, b, c, d\}$

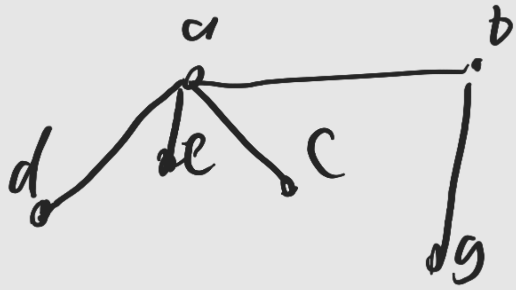
The next vector is e

$Q = [b, c, d, e]$ visited $\{a, b, c, d, e\}$

$T =$ 

All the neighbors of a have now been visited, so we can take the first vertex in Q, b, and start from there. The first neighbor of b that has not been visited is g

$Q = [c, d, e, g]$

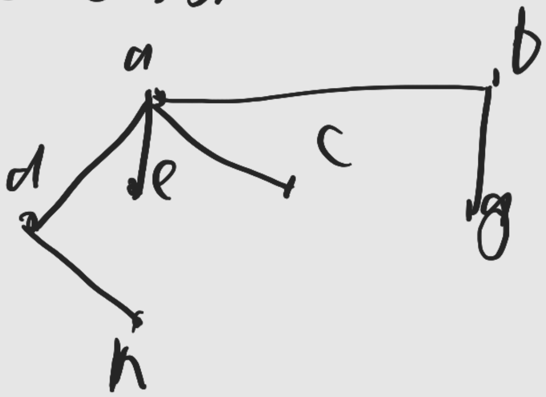
$T =$ 

visited = {a, b, c, d, e, g

all neighbors of b have been visited, so we pick now c from Q, all its neighbours have been visited, so the next one is d:

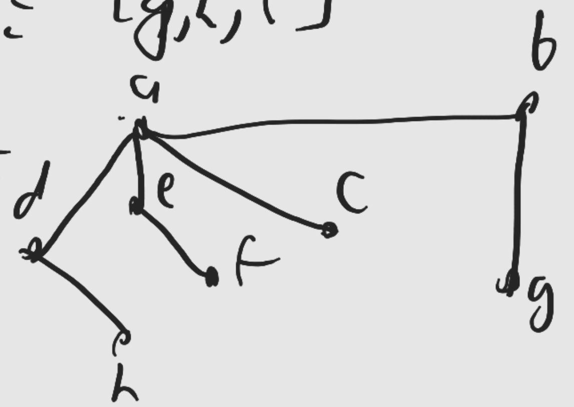
from d we can go to h:

$Q = [e, g, h]$

$T =$ 

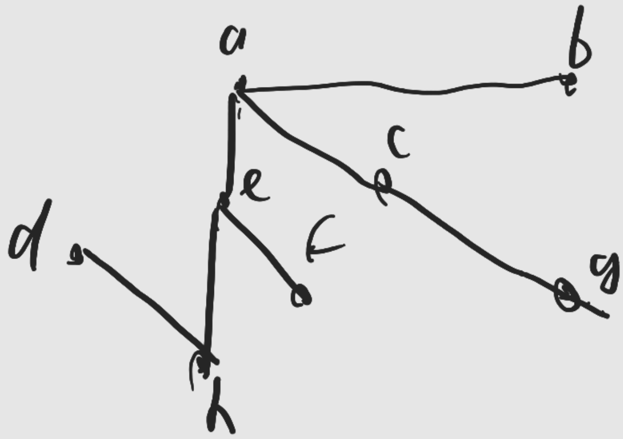
The next step gives

$Q = [g, h, f]$

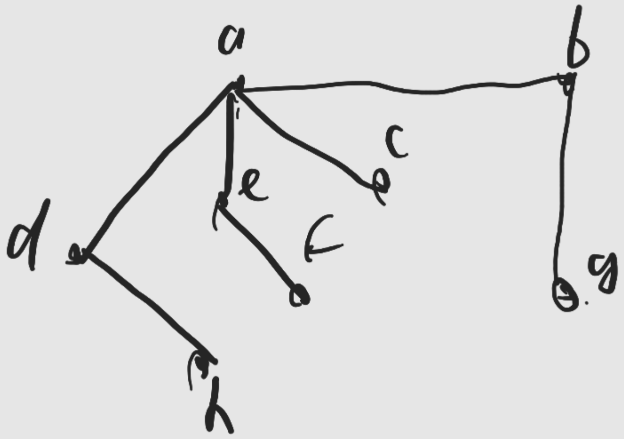
$T =$ 

This is the spanning tree.

ii)

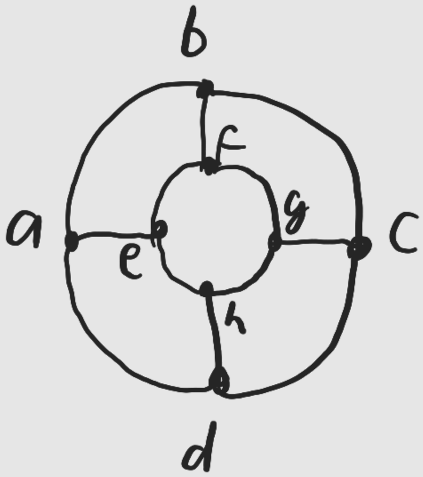


iii)



order: a, b, c, d,
h, g, f, e

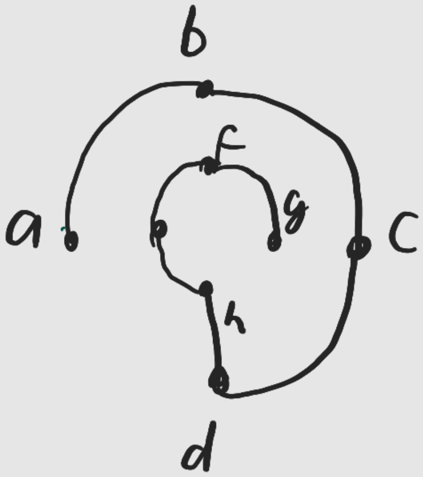
b) For the graph



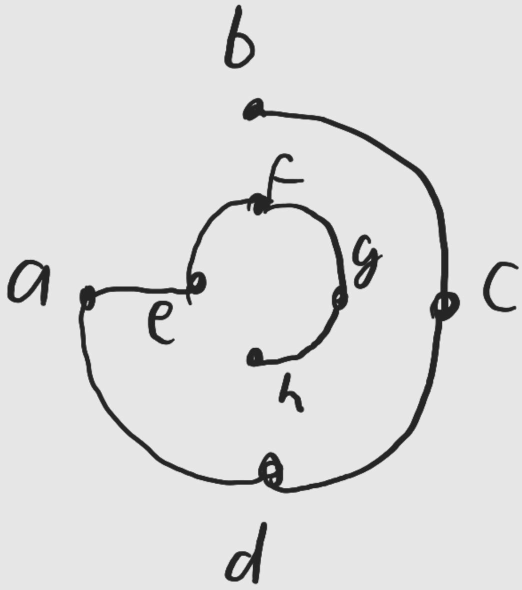
we have:

depth-first

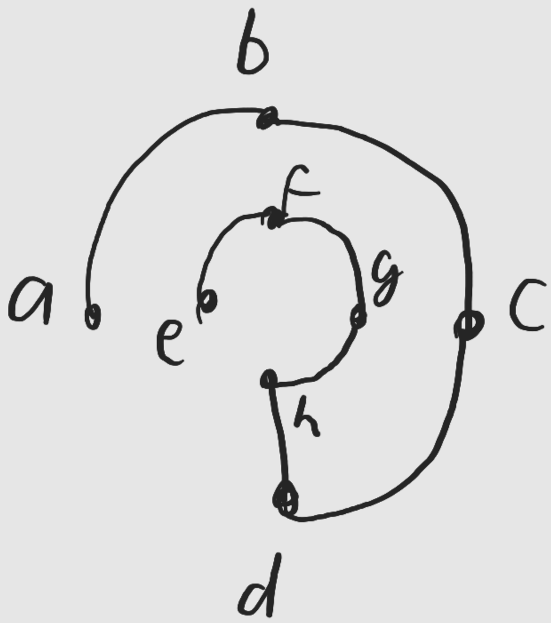
i)



ii)

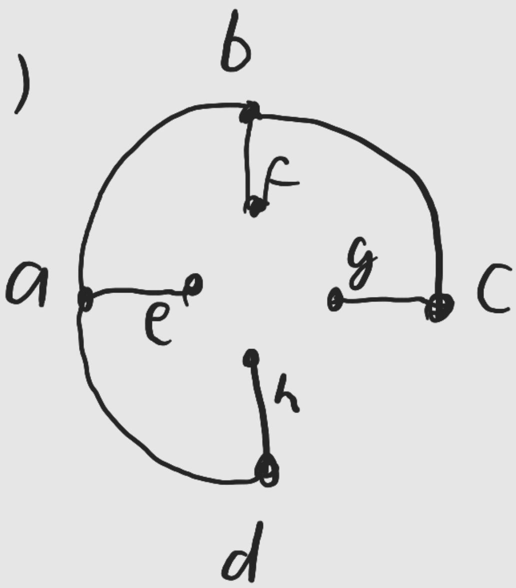


iii)

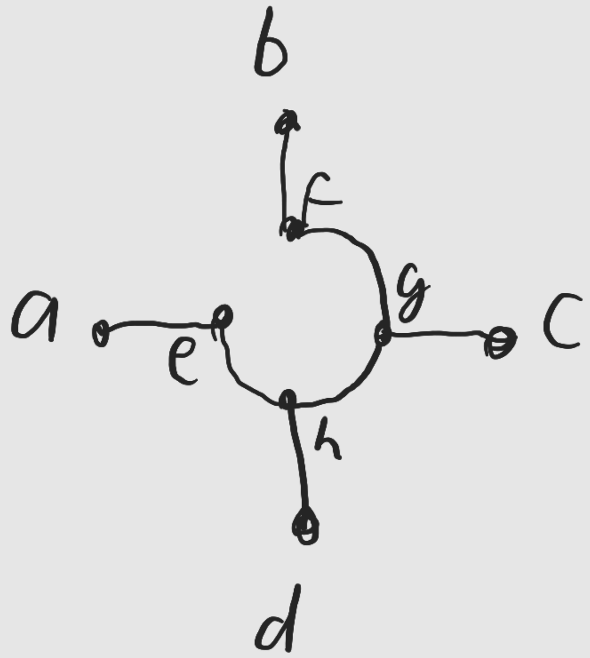


Breadth-First:

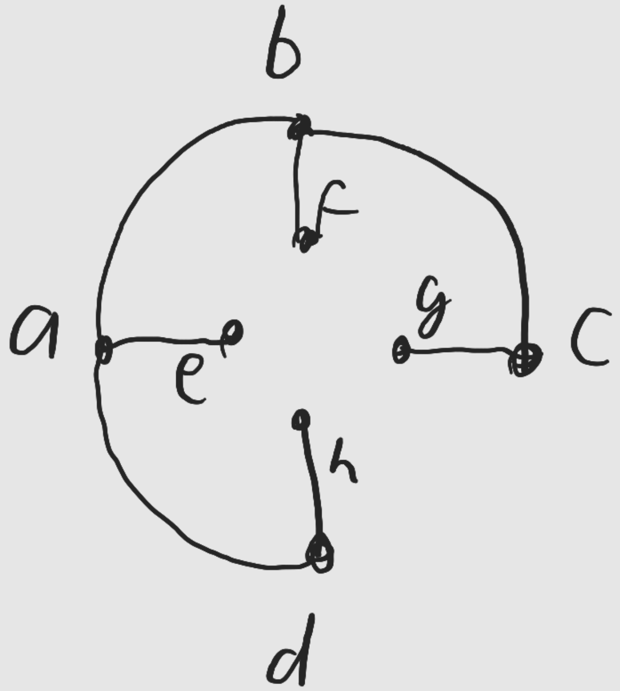
i)

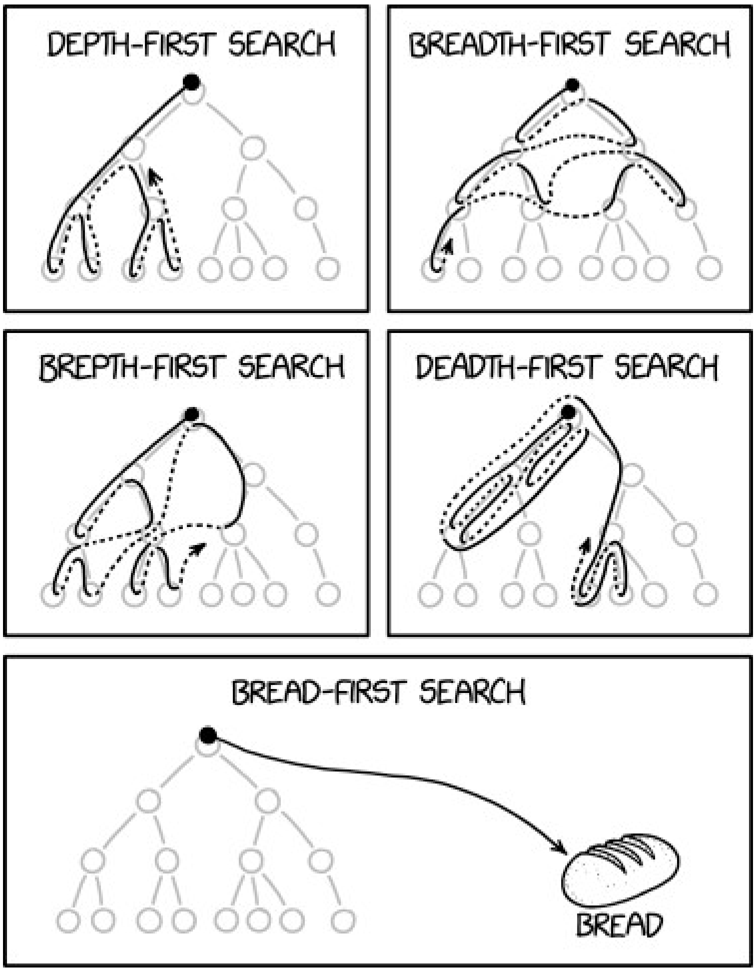


ii)



iii)





taken from: https://xkcd.com/2407/