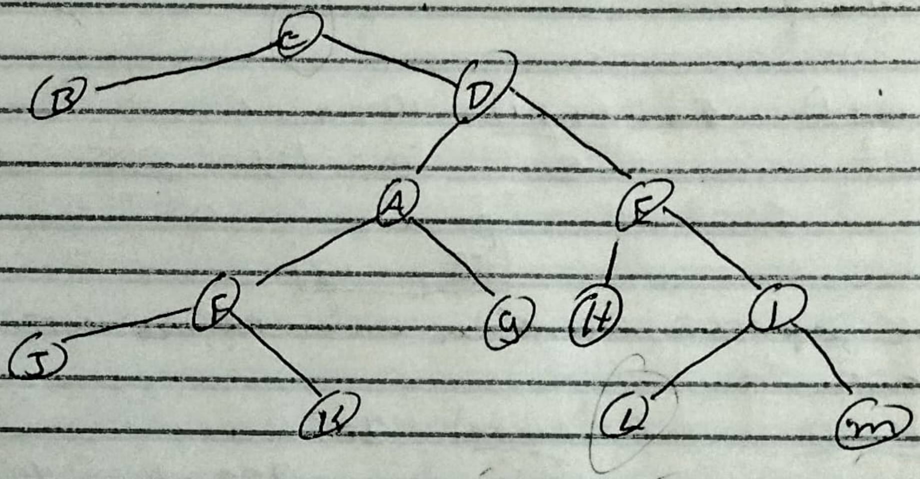
Exercício Parte 1 - Aula 8

Execute a busca em profundidade visando encontrar o vértice L



V[C]: $\underline{B}$, D

V(B) = $\underline{C}$

V(D] = A, $\overline{C}$, E

V(A] = $\overline{D}$, G, F

V[F] = $\overline{A}$, $\overline{J}$, $\overline{K}$

V(J] = $\overline{F}$, K

V[K]: $\overline{F}$

V(g] = $\overline{A}$

V(E]: $\overline{D}$, $\overline{I}$, H

V(I] = $\overline{L}$, $\overline{E}$, M

V(L] = $\overline{I}$

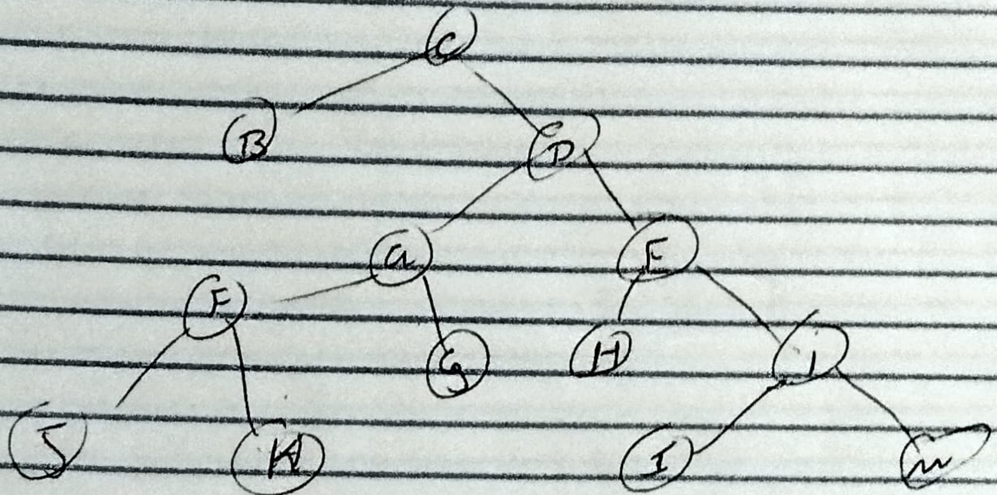
V(m] = I

V(H) = E

V[C) -> V(B] -> V(D) - V(A]
V(F] -> V(J] -> V(K] -> V(G)
V(E] -> V(H) -> V(I] -> V(L]
V

Exercício - Parte 2

1) Execute a busca em largura visando encontrar o vértice L:



Visitando = C, B, D, A, E, F, G, H, I, J, K, L
Fila = B, D, A, E, F, G, H, I, J, K, L, M

② Escreva um pseudocógico para busca em largura

```
class node {
    int value;
    List<node> vizinho
}
  public void BFS(Node start){
    Queue<node> queue = new List<>();
    start.visited = True;
    Queue.add(start)
    while(!queue.isEmpty()){
        Node current = queue.remove()
      For(Node Vizinho : current.vizinho){
          IF(!vizinho.visited){
              vizinho.visited = True
              queue.add(vizinho);
```