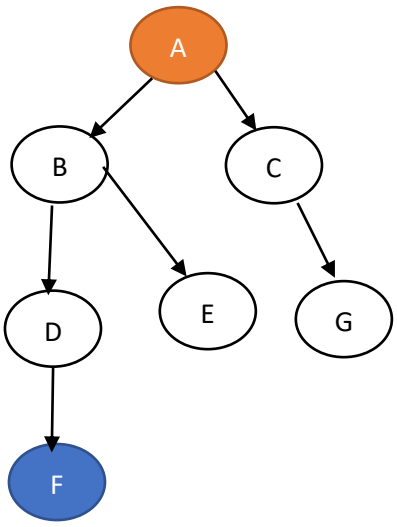
Les Algorithmes de résolution de problème

Exemple



Profondeur

1 O={A}
F={}
O={B,C}
F={A}
O={D,E,C}
F={A,B}
O={F,E,C}
F={A,B,D}

2 O={A}
F={}
O={C,B}
F={A}
O={G,B}
F={A,C}
O={B}
F={A,C}
O={B}
F={A}
O={D,E,}
F={A,B}
O={F,E,}
F={A,B,D}

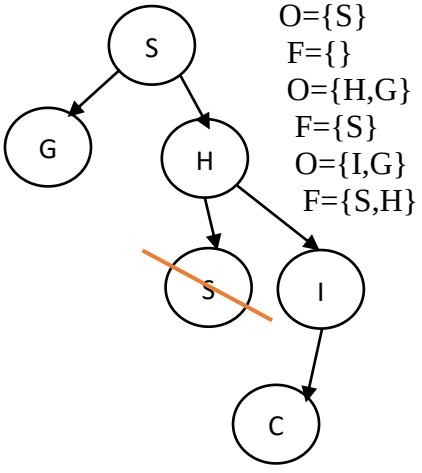
Largeur

O={A}
F={}
O={B,C}
F={A}
O={C,D,E}
F={A,B}
O={D,E,G}
F={A,B,C}
O={E,G,F}
F={A,B,C,D}

Exercice 1 TD 2

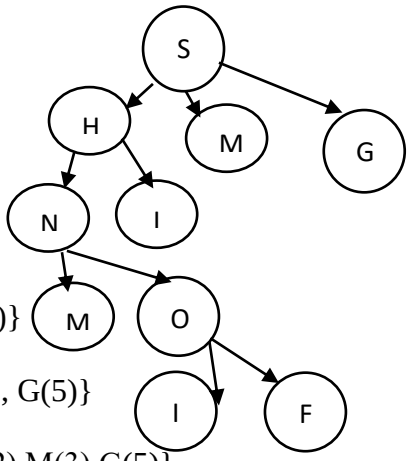
			A	B	
			C	D	E
G	S	H	I	J	K
L	M	N	O	F	

Algo en profondeur

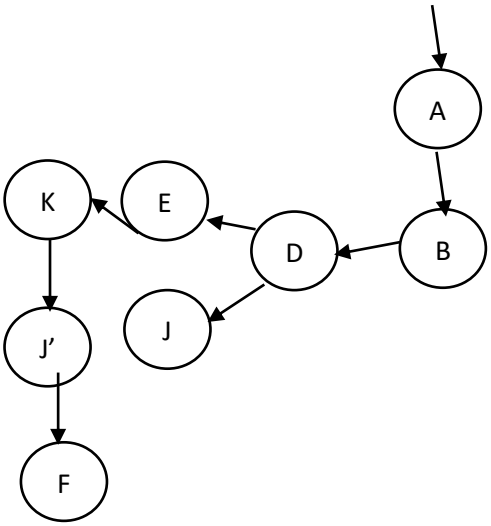


O={S}
F={}
O={H,G}
F={S}
O={I,G}
F={S,H}

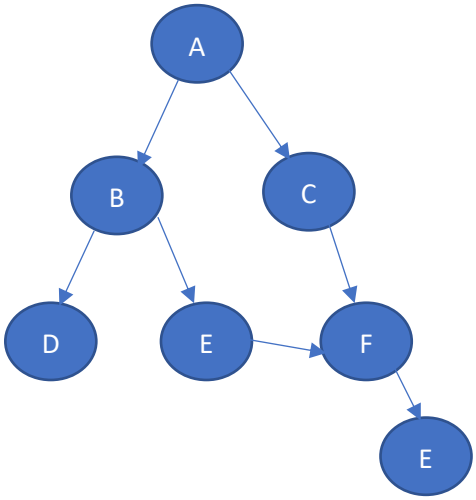
Hill Climbing



O={S(4)}
F={}
O={H(3), M(3),G(5)}
F={S(4)}
O={N(2), I(2),M(3), G(5)}
F={S(4), H(3)}
O={o(1), M'(3),I(2),M(3),G(5)}
F={S,H,N}



O={C,G}
F={S,H,I}
O={A,G}
F={S,H,I,C}
O={B,G}
F={S,H,I,C,A}
O={D,G}
F={S,H,I,C,A,B}
O={E,J,G}
F={S,H,I,C,A,B,D}
O={K,J,G}
F={S,H,I,C,A,B,D,E}
O={J',j,G}
F={ S,H,I,C,A,B,D,E, K}
O={F,j,G}
F={ S,H,I,C,A,B,D,E, K,J'}



L'algorithme de recherche en profondeur

```
From queue import PriorityQueue
Graph = {'A' : set (['B','C']),
                'B' :set(['D','E']),
                 'C' :set (['F']),
                  'E' :set (['F']),
                   'F' :set (['E'])}
Def dfs(graph, start, goal)
Visited= []
Closed =[]
Queue = PriorityQueue()
Queue.put(0,start,closed, visited)
 While not queue.empty()
        Depth, current_node, closed, visited = queue.get()
        If current_node == goal
                Return closed + [current_node]
        Visited = visited + [current_node]
        Child_nodes = graph[current_node]
        For node in child_nodes
                If node == goal
                        Return closed + [node]
                Depth = len(closed)
                Queue.put(depth, node,closed +[node],
visited)

 Return Closed
Closed =dfs(graph, A ,E)
Print(« closed »,closed)
```