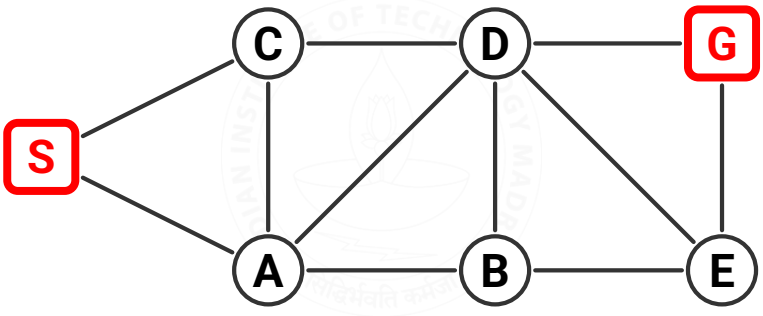
Practice Assignment: Example 1
Depth First Iterative Deepening (DFID-N)

Prepared by S. Baskaran

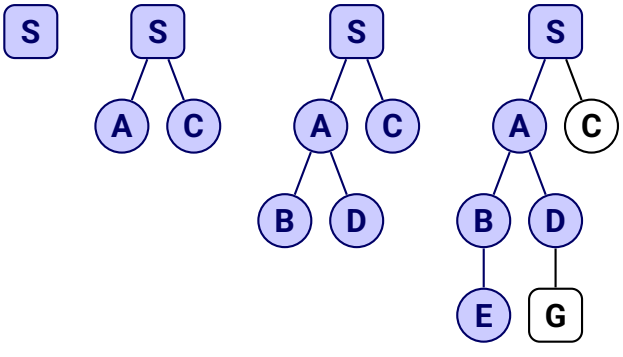
State Space



MoveGen returns nodes in ALPHABETICAL order.

```
S -> A,C
A -> B,C,D,S
B -> A,D,E
C -> A,D,S
D -> A,B,C,E,G
E -> B,D,G
G -> D,E
```

DFID-N Search Trees (d=0,1,2,3)



DFID-N Solution

The solution provided below is based on the DFID-N algorithm published in the Week 2 Notes.

```
OPEN and CLOSED carry triples: (NODE,PARENT,DEPTH)

***[DEPTH BOUND = 0]***

OPEN        (S,null,0):[]
CLOSED      []

1.
NODE        S
close       (S,null,0)

OPEN        []
CLOSED      (S,null,0):[]


***[DEPTH BOUND = 1]***

OPEN        (S,null,0):[]
CLOSED      []

1.
NODE        S
close       (S,null,0)
moveGen     A:C:[]
newNodes    A:C:[]
newPairs    (A,S,1):(C,S,1):[]

OPEN        (A,S,1):(C,S,1):[]
CLOSED      (S,null,0):[]

2.
NODE        A
close       (A,S,1)
moveGen     B:C:D:S:[]
newNodes    B:D:[]
newPairs    (B,A,2):(D,A,2):[]

OPEN        (B,A,2):(D,A,2):(C,S,1):[]
CLOSED      (A,S,1):(S,null,0):[]

3.
NODE        C
close       (C,S,1)

OPEN        []
CLOSED      (C,S,1):(A,S,1):(S,null,0):[]


***[DEPTH BOUND = 2]***

OPEN        (S,null,0):[]
CLOSED      []

1.
NODE        S
close       (S,null,0)
moveGen     A:C:[]
newNodes    A:C:[]
newPairs    (A,S,1):(C,S,1):[]

OPEN        (A,S,1):(C,S,1):[]
CLOSED      (S,null,0):[]

2.
NODE        A
close       (A,S,1)
moveGen     B:C:D:S:[]
newNodes    B:D:[]
newPairs    (B,A,2):(D,A,2):[]

OPEN        (B,A,2):(D,A,2):(C,S,1):[]
CLOSED      (A,S,1):(S,null,0):[]

3.
NODE        B
close       (B,A,2)

OPEN        (D,A,2):(C,S,1):[]
CLOSED      (B,A,2):(A,S,1):(S,null,0):[]

4.
NODE        D
close       (D,A,2)

OPEN        (C,S,1):[]
CLOSED      (D,A,2):(B,A,2):(A,S,1):(S,null,0):[]

5.
NODE        C
close       (C,S,1)
moveGen     A:D:S:[]
newNodes    []
newPairs    []

OPEN        []
CLOSED      (C,S,1):(D,A,2):(B,A,2):(A,S,1):(S,null,0):[]


***[DEPTH BOUND = 3]***

OPEN        (S,null,0):[]
CLOSED      []

1.
NODE        S
close       (S,null,0)
moveGen     A:C:[]
newNodes    A:C:[]
newPairs    (A,S,1):(C,S,1):[]

OPEN        (A,S,1):(C,S,1):[]
CLOSED      (S,null,0):[]

2.
NODE        A
close       (A,S,1)
moveGen     B:C:D:S:[]
newNodes    B:D:[]
newPairs    (B,A,2):(D,A,2):[]

OPEN        (B,A,2):(D,A,2):(C,S,1):[]
CLOSED      (A,S,1):(S,null,0):[]

3.
NODE        B
close       (B,A,2)
moveGen     A:D:E:[]
newNodes    E:[]
newPairs    (E,B,3):[]

OPEN        (E,B,3):(D,A,2):(C,S,1):[]
CLOSED      (B,A,2):(A,S,1):(S,null,0):[]

4.
NODE        E
close       (E,B,3)

OPEN        (D,A,2):(C,S,1):[]
CLOSED      (E,B,3):(B,A,2):(A,S,1):(S,null,0):[]

5.
NODE        D
close       (D,A,2)
moveGen     A:B:C:E:G:[]
newNodes    G:[]
newPairs    (G,D,3):[]

OPEN        (G,D,3):(C,S,1):[]
CLOSED      (D,A,2):(E,B,3):(B,A,2):(A,S,1):(S,null,0):[]

6.
NODE        G
GOAL        G

OPEN        (G,D,3):(C,S,1):[]
CLOSED      (D,A,2):(E,B,3):(B,A,2):(A,S,1):(S,null,0):[]

PATH        S:A:D:G:[]
```