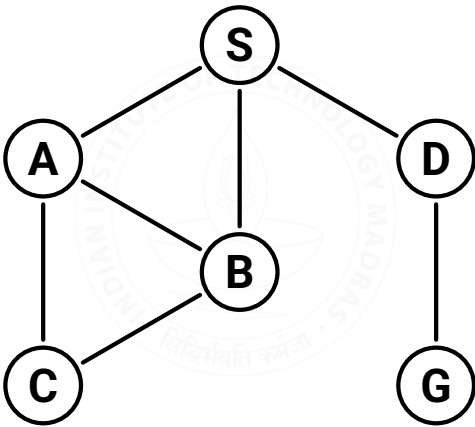


Lecture Example 2

Breadth First Search: Cases 1, 2 and 3

Prepared by S. Baskaran

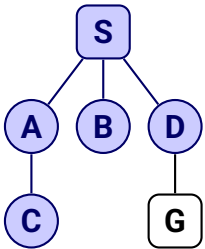
State Space



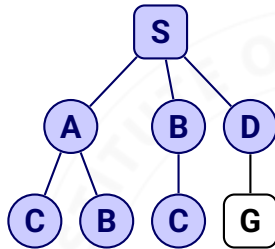
X	MoveGen(X)
S	[A, B, D]
A	[C, B, S]
B	[S, A, C]
C	[B, A]
D	[S, G]
G	[D]

Breadth First Search – Search Trees

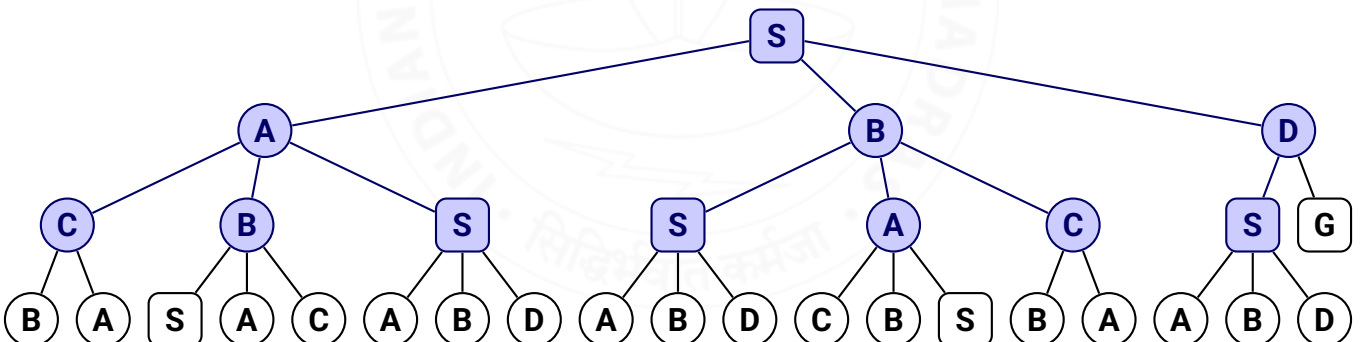
BFS Case 1



BFS Case 2



BFS Case 3



Breadth First Search – Case 1

The solution for Case 1 is based on the BFS algorithm given in Week 2 Notes. For the Cases 2 and 3, take BFS Case 1 and make suitable modifications according to what is discussed in the lecture. Try it yourself.

BFS Case 1 removes both OPEN nodes and CLOSED nodes from the output of MoveGen.

OPEN and CLOSED carry pairs: (NODE,PARENT)

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

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

OPEN (A,S):(B,S):(D,S):[]
CLOSED (S,null):[]

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

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

3.
NODE B
close (B,S)
moveGen S:A:C:[]
newNodes []
newPairs []

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

4.
NODE D
close (D,S)
moveGen S:G:[]
newNodes G:[]
newPairs (G,D):[]

OPEN (C,A):(G,D):[]
CLOSED (D,S):(B,S):(A,S):(S,null):[]

5.
NODE C
close (C,A)
moveGen B:A:[]
newNodes []
newPairs []

OPEN (G,D):[]
CLOSED (C,A):(D,S):(B,S):(A,S):(S,null):[]

6.
NODE G
GOAL G

OPEN (G,D):[]
CLOSED (C,A):(D,S):(B,S):(A,S):(S,null):[]

PATH S:D:G:[]

Breadth First Search – Case 2

The solution for Case 1 is based on the BFS algorithm given in Week 2 Notes. For the Cases 2 and 3, take BFS Case 1 and make suitable modifications according to what is discussed in the lecture. Try it yourself.

BFS Case 2 removes only the CLOSED nodes from the output of MoveGen.

OPEN and CLOSED carry pairs: (NODE,PARENT)

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

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

OPEN (A,S):(B,S):(D,S):[]
CLOSED (S,null):[]

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

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

3.
NODE B
close (B,S)
moveGen S:A:C:[]
newNodes C:[]
newPairs (C,B):[]

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

4.
NODE D
close (D,S)
moveGen S:G:[]
newNodes G:[]
newPairs (G,D):[]

OPEN (C,A):(B,A):(C,B):(G,D):[]
CLOSED (D,S):(B,S):(A,S):(S,null):[]

5.
NODE C
close (C,A)
moveGen B:A:[]
newNodes []
newPairs []

OPEN (B,A):(C,B):(G,D):[]
CLOSED (C,A):(D,S):(B,S):(A,S):(S,null):[]

6.
NODE B
close (B,A)
moveGen S:A:C:[]
newNodes []
newPairs []

OPEN (C,B):(G,D):[]
CLOSED (B,A):(C,A):(D,S):(B,S):(A,S):(S,null):[]

7.
NODE C
close (C,B)
moveGen B:A:[]
newNodes []
newPairs []

OPEN (G,D):[]
CLOSED (C,B):(B,A):(C,A):(D,S):(B,S):(A,S):(S,null):[]

8.
NODE G
GOAL G

OPEN (G,D):[]
CLOSED (C,B):(B,A):(C,A):(D,S):(B,S):(A,S):(S,null):[]

PATH S:D:G:[]

Breadth First Search – Case 3

The solution for Case 1 is based on the BFS algorithm given in Week 2 Notes. For the Cases 2 and 3, take BFS Case 1 and make suitable modifications according to what is discussed in the lecture. Try it yourself.

BFS Case 3 does not remove any nodes from the output of MoveGen.

OPEN and CLOSED carry pairs: (NODE,PARENT)

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

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

OPEN (A,S):(B,S):(D,S):[]
CLOSED (S,null):[]

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

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

3.
NODE B
close (B,S)
moveGen S:A:C:[]
newNodes S:A:C:[]
newPairs (S,B):(A,B):(C,B):[]

OPEN (D,S):(C,A):(B,A):(S,A):(S,B):
(A,B):(C,B):[]

CLOSED (B,S):(A,S):(S,null):[]

4.
NODE D
close (D,S)
moveGen S:G:[]
newNodes S:G:[]
newPairs (S,D):(G,D):[]

OPEN (C,A):(B,A):(S,A):(S,B):(A,B):
(C,B):(S,D):(G,D):[]

CLOSED (D,S):(B,S):(A,S):(S,null):[]

5.
NODE C
close (C,A)
moveGen B:A:[]
newNodes B:A:[]
newPairs (B,C):(A,C):[]

OPEN (B,A):(S,A):(S,B):(A,B):(C,B):
(S,D):(G,D):(B,C):(A,C):[]

CLOSED (C,A):(D,S):(B,S):(A,S):(S,null):[]

6.
NODE B
close (B,A)
moveGen S:A:C:[]
newNodes S:A:C:[]
newPairs (S,B):(A,B):(C,B):[]

OPEN (S,A):(S,B):(A,B):(C,B):(S,D):
(G,D):(B,C):(A,C):(S,B):(A,B):
(C,B):[]

CLOSED (B,A):(C,A):(D,S):(B,S):(A,S):
(S,null):[]

7.
NODE S
close (S,A)
moveGen A:B:D:[]
newNodes A:B:D:[]
newPairs (A,S):(B,S):(D,S):[]

OPEN (S,B):(A,B):(C,B):(S,D):(G,D):
(B,C):(A,C):(S,B):(A,B):(C,B):
(A,S):(B,S):(D,S):[]

CLOSED (S,A):(B,A):(C,A):(D,S):(B,S):
(A,S):(S,null):[]

8.
NODE S
close (S,B)
moveGen A:B:D:[]
newNodes A:B:D:[]
newPairs (A,S):(B,S):(D,S):[]

OPEN (A,B):(C,B):(S,D):(G,D):(B,C):
(A,C):(S,B):(A,B):(C,B):(A,S):
(B,S):(D,S):(A,S):(B,S):(D,S):[]

CLOSED (S,B):(S,A):(B,A):(C,A):(D,S):
(B,S):(A,S):(S,null):[]

9.
NODE A
close (A,B)
moveGen C:B:S:[]
newNodes C:B:S:[]
newPairs (C,A):(B,A):(S,A):[]

OPEN (C,B):(S,D):(G,D):(B,C):(A,C):
(S,B):(A,B):(C,B):(A,S):(B,S):
(D,S):(A,S):(B,S):(D,S):(C,A):
(B,A):(S,A):[]

CLOSED (A,B):(S,B):(S,A):(B,A):(C,A):
(D,S):(B,S):(A,S):(S,null):[]

10.
NODE C
close (C,B)
moveGen B:A:[]
newNodes B:A:[]
newPairs (B,C):(A,C):[]

OPEN (S,D):(G,D):(B,C):(A,C):(S,B):
(A,B):(C,B):(A,S):(B,S):(D,S):
(A,S):(B,S):(D,S):(C,A):(B,A):
(S,A):(B,C):(A,C):[]

CLOSED (C,B):(A,B):(S,B):(S,A):(B,A):
(C,A):(D,S):(B,S):(A,S):(S,null):[]

11.
NODE S
close (S,D)
moveGen A:B:D:[]
newNodes A:B:D:[]
newPairs (A,S):(B,S):(D,S):[]

OPEN (G,D):(B,C):(A,C):(S,B):(A,B):
(C,B):(A,S):(B,S):(D,S):(A,S):
(B,S):(D,S):(C,A):(B,A):(S,A):
(B,C):(A,C):(A,S):(B,S):(D,S):[]

CLOSED (S,D):(C,B):(A,B):(S,B):(S,A):
(B,A):(C,A):(D,S):(B,S):(A,S):
(S,null):[]

12.
NODE G
GOAL G

OPEN (G,D):(B,C):(A,C):(S,B):(A,B):
(C,B):(A,S):(B,S):(D,S):(A,S):
(B,S):(D,S):(C,A):(B,A):(S,A):
(B,C):(A,C):(A,S):(B,S):(D,S):[]

CLOSED (S,D):(C,B):(A,B):(S,B):(S,A):
(B,A):(C,A):(D,S):(B,S):(A,S):
(S,null):[]

PATH S:D:G:[]