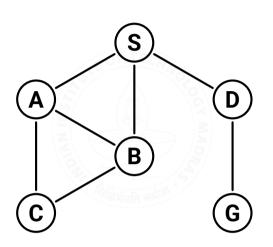
Lecture Example 2

Breadth First Search: Cases 1, 2 and 3

Prepared by S. Baskaran

State Space

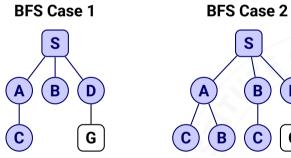


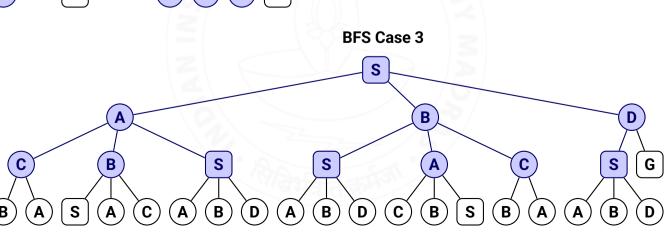
Χ	MoveGen(X)
S	[A,B,D]
Α	[C,B,S]
В	[S,A,C]
C	[B,A]
D	[S,G]
G	[D]

Breadth First Search - Search Trees

S

В





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)
            (S, null):[]
OPEN
CLOSED
            []
1.
NODE
            S
            (S, null)
close
moveGen
            A:B:D:[]
newNodes
            A:B:D:[]
newPairs
            (A,S):(B,S):(D,S):[]
            (A,S):(B,S):(D,S):[]
OPEN
CLOSED
            (S, null):[]
2.
NODE
            Α
            (A,S)
close
            C:B:S:[]
moveGen
newNodes
            C:[]
            (C,A):[]
newPairs
            (B,S):(D,S):(C,A):[]
OPEN
CLOSED
            (A,S):(S,null):[]
3.
NODE
            В
close
            (B,S)
            S:A:C:[]
moveGen
            []
newNodes
newPairs
            []
OPEN
            (D,S):(C,A):[]
            (B,S):(A,S):(S,null):[]
CLOSED
4.
NODE
            D
close
            (D,S)
            S:G:[]
moveGen
            G:[]
newNodes
newPairs
            (G,D):[]
OPEN
            (C, A):(G, D):[]
            (D,S):(B,S):(A,S):(S,null):[]
CLOSED
5.
NODE
            С
            (C, A)
close
            B:A:[]
moveGen
            []
newNodes
            []
newPairs
OPEN
            (G,D):[]
            (C,A):(D,S):(B,S):(A,S):(S,null):[]
CLOSED
6.
NODE
            G
            G
GOAL
            (G,D):[]
OPEN
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)
            A:B:D:[]
moveGen
newNodes
            A:B:D:[]
newPairs
            (A,S):(B,S):(D,S):[]
OPEN
            (A,S):(B,S):(D,S):[]
            (S, null):[]
CLOSED
2.
NODE
            Α
            (A,S)
close
            C:B:S:[]
moveGen
            C:B:[]
newNodes
            (C,A):(B,A):[]
newPairs
OPEN
            (B,S):(D,S):(C,A):(B,A):[]
CLOSED
            (A,S):(S,null):[]
3.
NODE
            В
            (B,S)
close
            S:A:C:[]
moveGen
            C:[]
newNodes
newPairs
            (C,B):[]
OPEN
            (D,S):(C,A):(B,A):(C,B):[]
            (B,S):(A,S):(S,null):[]
CLOSED
4.
NODE
            D
close
            (D,S)
moveGen
            S:G:[]
            G:[]
newNodes
            (G,D):[]
newPairs
            (C, A):(B, A):(C, B):(G, D):[]
OPEN
CLOSED
            (D,S):(B,S):(A,S):(S,null):[]
5.
NODE
            С
close
            (C,A)
            B:A:[]
moveGen
            []
newNodes
            []
newPairs
OPEN
            (B, A):(C, B):(G, D):[]
            (C,A):(D,S):(B,S):(A,S):(S,null):[]
CLOSED
6.
NODE
            В
            (B,A)
close
            S:A:C:[]
moveGen
newNodes
            [\ ]
newPairs
            []
OPEN
            (C,B):(G,D):[]
            (B,A):(C,A):(D,S):(B,S):(A,S):
CLOSED
            (S, null):[]
7.
NODE
            С
            (C,B)
close
moveGen
            B:A:[]
            []
newNodes
            []
newPairs
OPEN
            (G,D):[]
            (C,B):(B,A):(C,A):(D,S):(B,S):
CLOSED
            (A,S):(S,null):[]
8.
NODE
            G
            G
GOAL
OPEN
            (G,D):[]
            (C,B):(B,A):(C,A):(D,S):(B,S):
CLOSED
            (A,S):(S,null):[]
```

S:D:G:[]

PATH

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.

(S, null):[]

OPEN

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

OPEN and CLOSED carry pairs: (NODE, PARENT)

Breadth First Search – Case 3

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

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

CLOSED 4. NODE close moveGen newNodes newPairs **OPEN** CLOSED 5. NODE close moveGen 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):[] (C,A):(D,S):(B,S):(A,S):(S,null):[] CLOSED 6. NODE В (B,A)close S:A:C:[] moveGen S:A:C:[] newNodes (S,B):(A,B):(C,B):[] newPairs **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 (S,A)close

A:B:D:[] moveGen A:B:D:[] newNodes (A,S):(B,S):(D,S):[] newPairs (S,B):(A,B):(C,B):(S,D):(G,D): OPEN (B,C):(A,C):(S,B):(A,B):(C,B): (A,S):(B,S):(D,S):[] (S,A):(B,A):(C,A):(D,S):(B,S): CLOSED (A,S):(S,null):[] 8. NODE S (S,B)close moveGen A:B:D:[] newNodes A:B:D:[] (A,S):(B,S):(D,S):[] newPairs **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):[] (S,B):(S,A):(B,A):(C,A):(D,S): CLOSED (B,S):(A,S):(S,null):[] 9. NODE Α close (A,B)C:B:S:[] moveGen 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):[] (A,B):(S,B):(S,A):(B,A):(C,A):CLOSED (D,S):(B,S):(A,S):(S,null):[] 10. NODE C (C,B) close B:A:[] moveGen B:A:[] newNodes 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. S NODE close (S,D)A:B:D:[] moveGen newNodes A:B:D:[] (A,S):(B,S):(D,S):[] newPairs

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):[]

> (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):[]

(S,D):(C,B):(A,B):(S,B):(S,A): (B,A):(C,A):(D,S):(B,S):(A,S):

12. NODE

GOAL

OPEN

CLOSED

PATH

G

G

(S, null):[]

S:D:G:[]