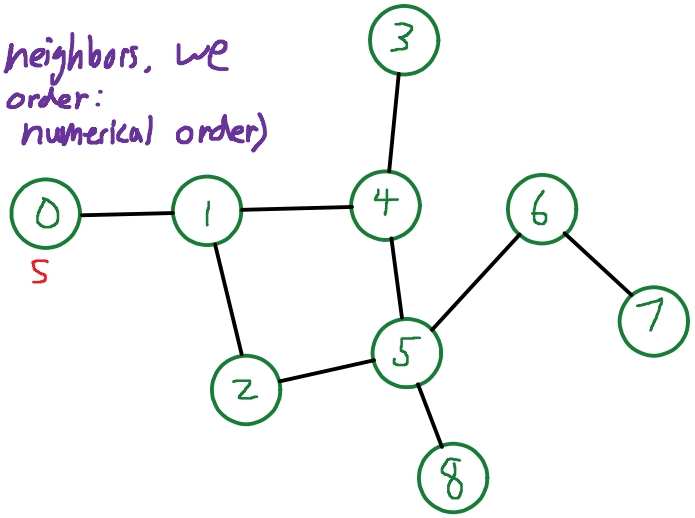
Depth First Search (DFS)

- Idea: Explore a graph by exploring as far as possible along each branch before backtracking
- How it works:
  1. Initialize:
     - Stack of nodes stk
     - Marked set marked (represents set of visited nodes)
     - Add first node to stk and marked
  2. Pop off topmost node v from stk
     If v not in marked:
     - Add v in preorder sequence
     - Add v to marked
     - for all adjacent nodes n of v not marked:
       add n to stk

     Else: continue
  3. End: Repeat 2 until stk empty
- Example: Asswme if we have 2+ neighbors, we want to visit each by numerical order:
  add neighbors to stack in reverse numerical order)



Stk:
[]

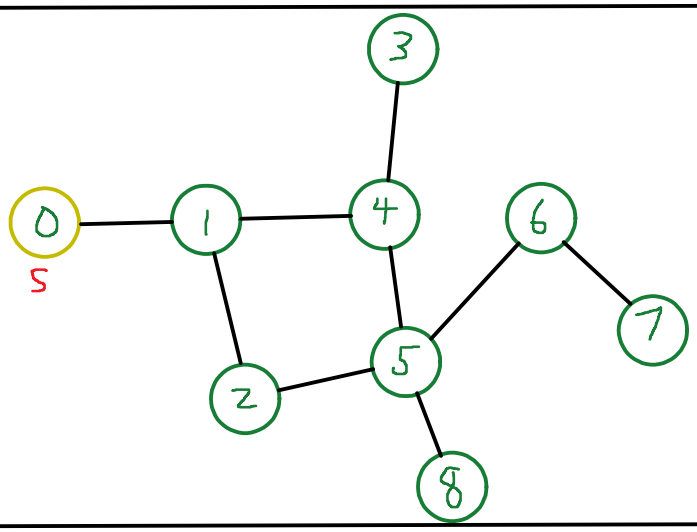
marked:
{}

Preorder:

Stk:
[0]

marked:
{0}

Preorder:
0

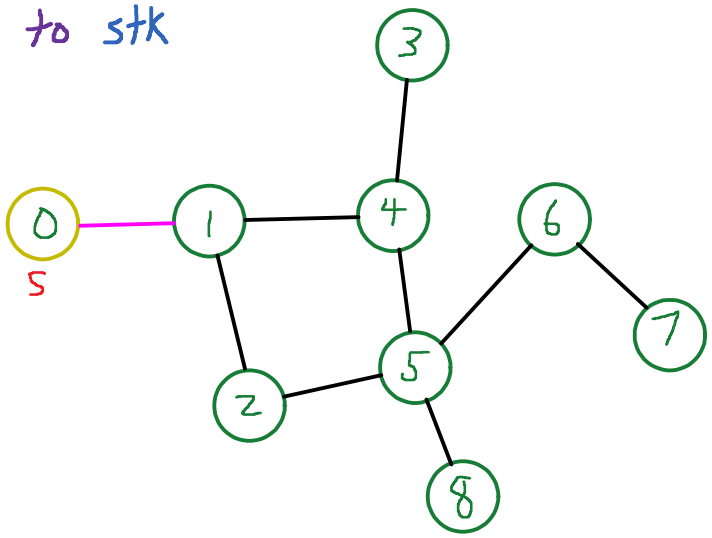


Pop off 0, add neighbors to stk

Stk:
[1]

marked:
{0}

PreOrder:
0

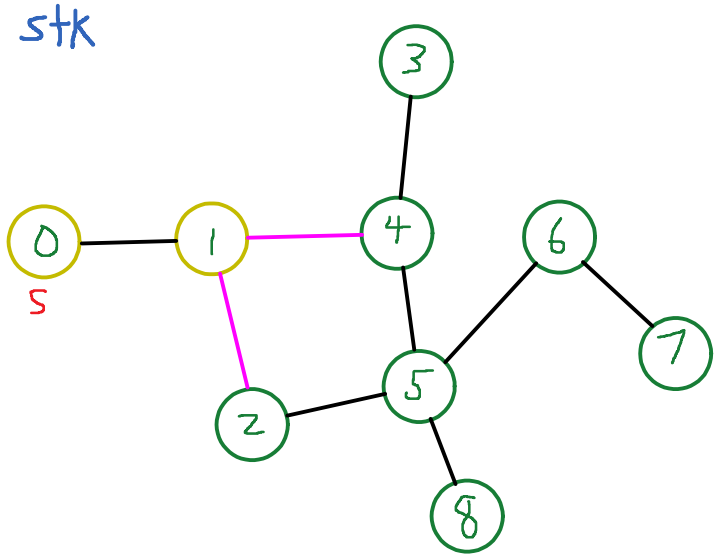


Pop off 1, add neighbors to stk

Stk: ← top of stack
[4 2]

marked:
{0 1}

PreOrder:
0 1

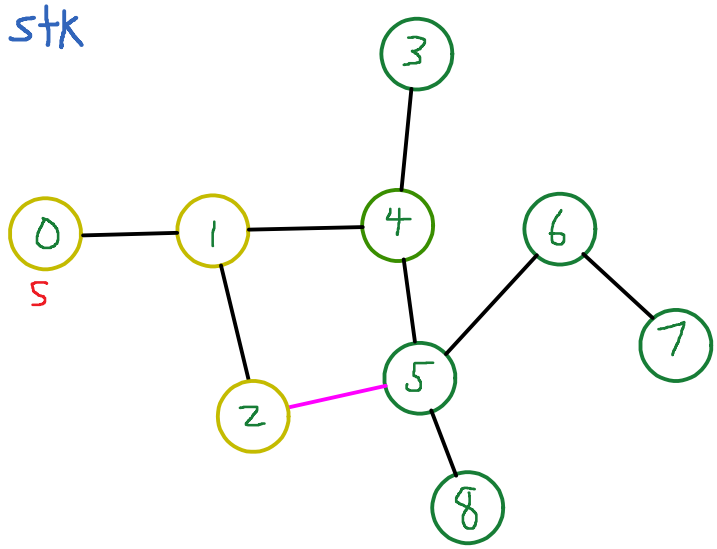


Pop off 2, add neighbors to stk

Stk
[4 5]

marked:
{0 1 2}

PreOrder:
0 1 2



Pop off 5, add neighbors to stk

Stk:

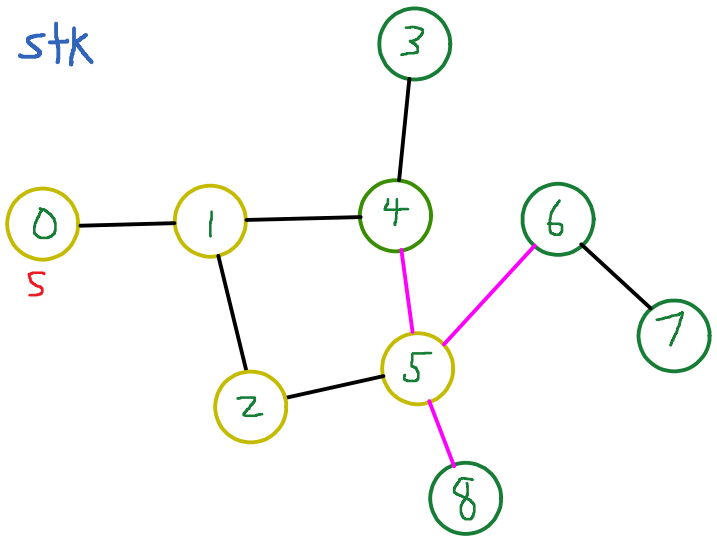
[4 8 6 4]

marked:

{0 1 2 5}

PreOrder:

0 1 2 5



Pop off 4, add neighbors to stk

Stk:

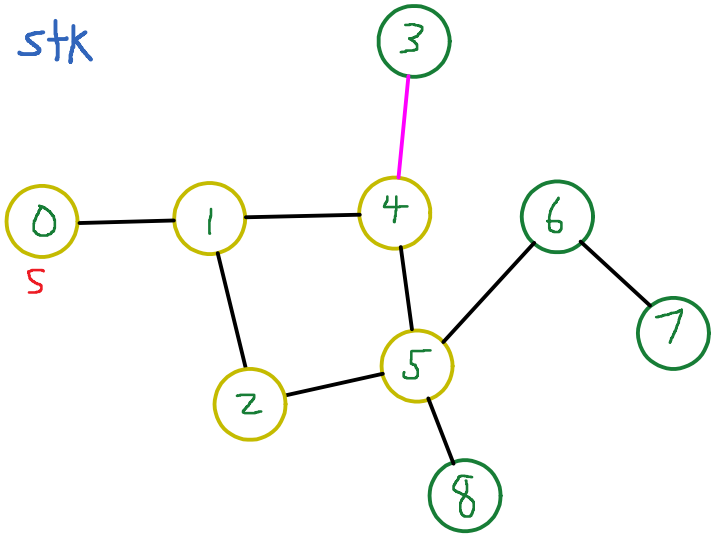
[4 8 6 3]

marked:

{0 1 2 5 4}

PreOrder:

0 1 2 5 4



Pop off 3, add neighbors to stk (none)

Stk:

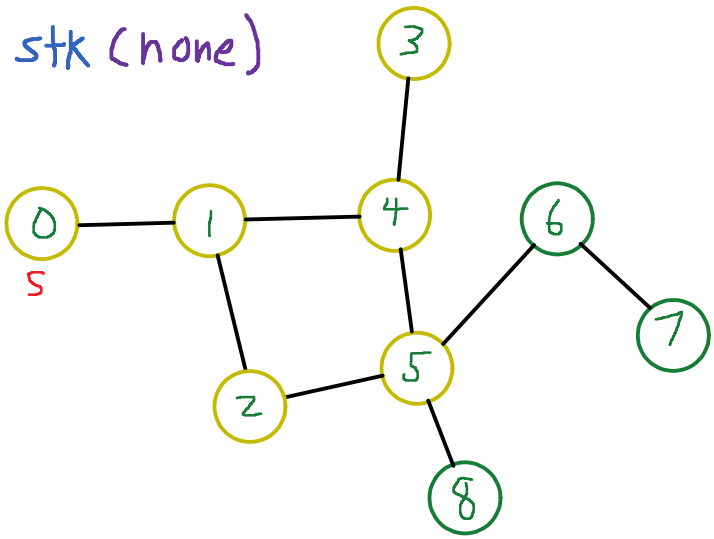
[4 8 6]

marked:

{0 1 2 5 4 3}

PreOrder:

0 1 2 5 4 3



Pop off 6, add neighbors to stk

Stk:

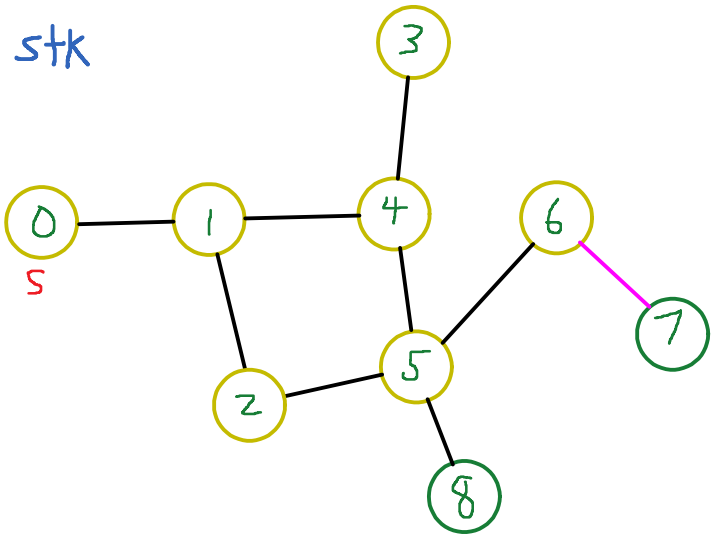
[4 8 7]

marked:

{0 1 2 5 4 3 6}

PreOrder:

0 1 2 5 4 3 6



Pop off 6, add neighbors to stk

Stk:

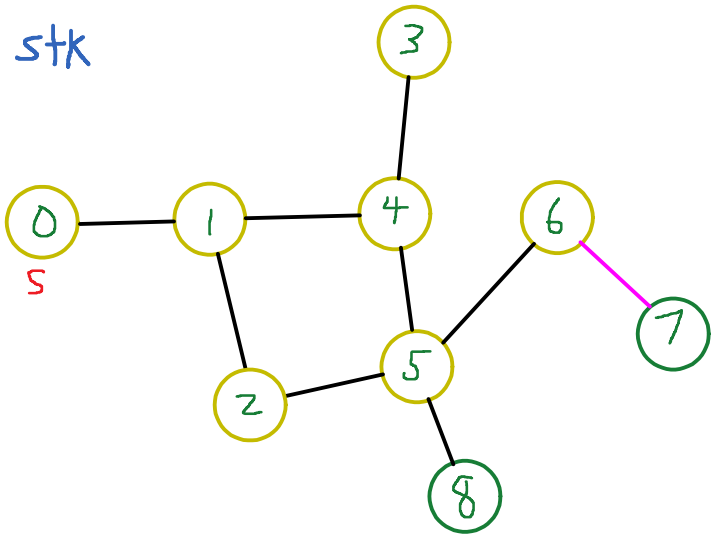
[4 8 7]

marked:

{0 1 2 5 4 3 6}

PreOrder:

0 1 2 5 4 3 6



Pop off 7, add neighbors to stk (none)

Stk:

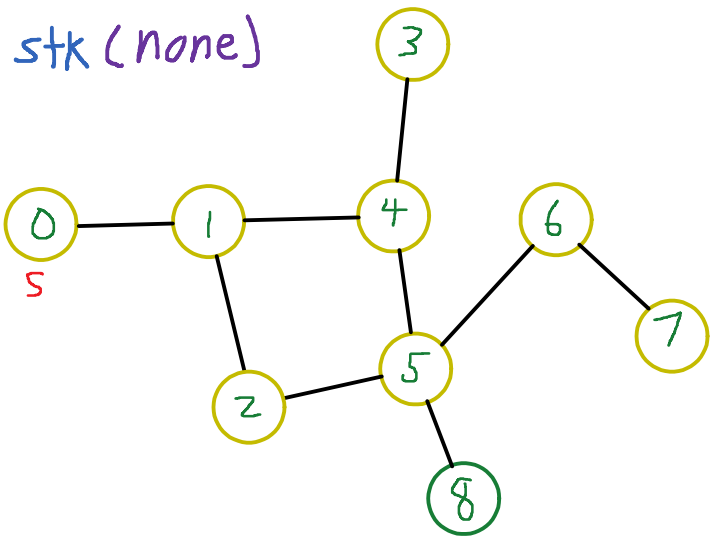
[4 8]

marked:

{0 1 2 5 4 3 6 7}

PreOrder:

0 1 2 5 4 3 6 7



Pop off 8, add neighbors to stk (none)

Stk:

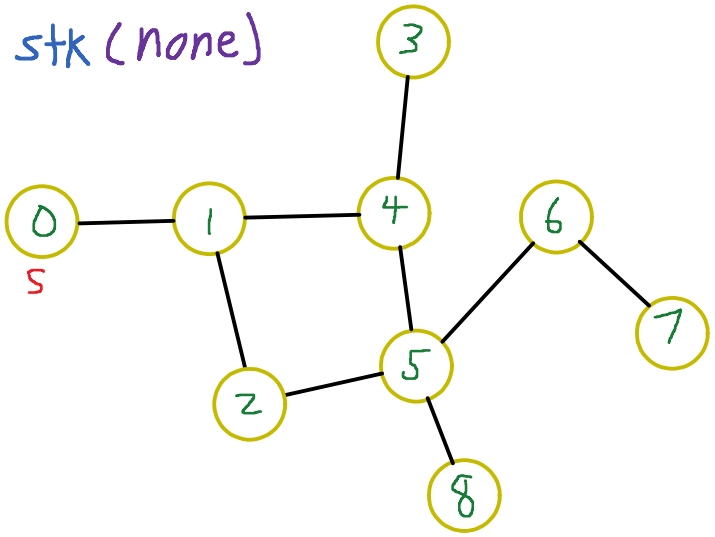
[4]

marked:

{0 1 2 5 4 3 6 7 8}

PreOrder:

0 1 2 5 4 3 6 7 8



Pop off 4, in marked so continue

Stk:

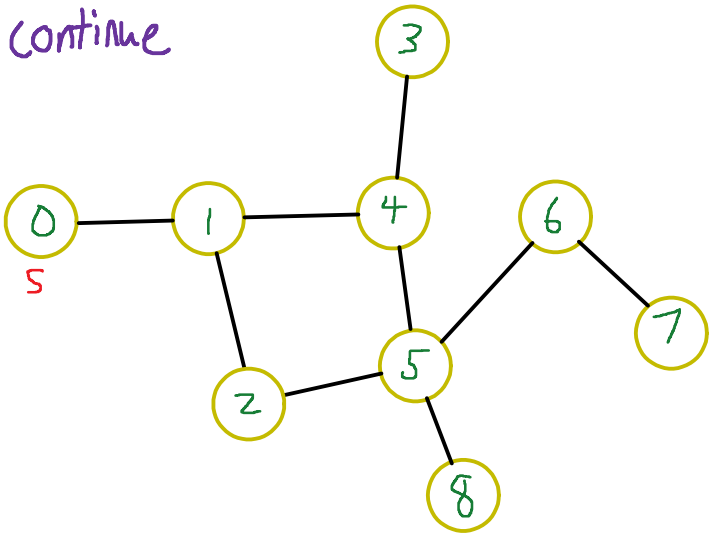
[]

marked:

{0 1 2 5 4 3 6 7 8}

PreOrder:

0 1 2 5 4 3 6 7 8



Stk empty, so done