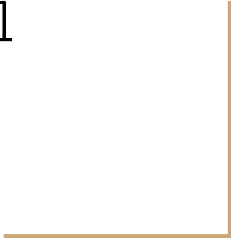


Programming, Problem Solving, and Algorithms

CPSC203, 2019 W1



Announcements

Project 2 is released. Due 11:59p, Nov 7.

“Problem of the Day” continues!

Today:

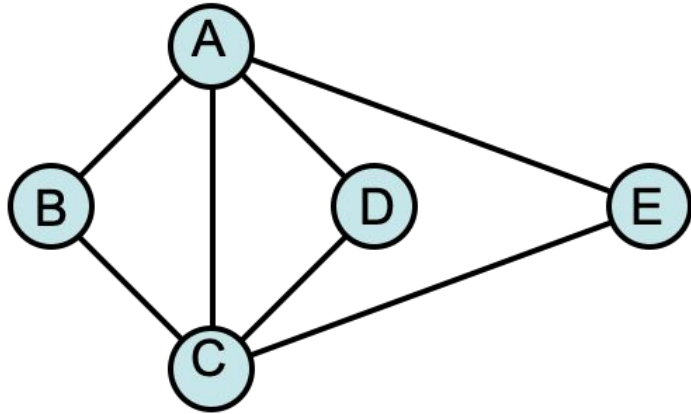
Markov Chains Fin

State Space Search

Representation

Implementation

Depth First Search



Algorithm DFS(G,v)

Input: graph G and start vertex v

Output: labeling of the edges of G in the connected component of v as discovery edges and back edges

setLabel(v , VISITED)

For all w in G .adjacentVertices(v)

if getLabel(w) = UNVISITED

 setLabel((v,w) ,DISCOVERY)

 DFS(G,w)

else if getLabel((v,w)) = UNEXPLORED

 setLabel(e ,BACK)

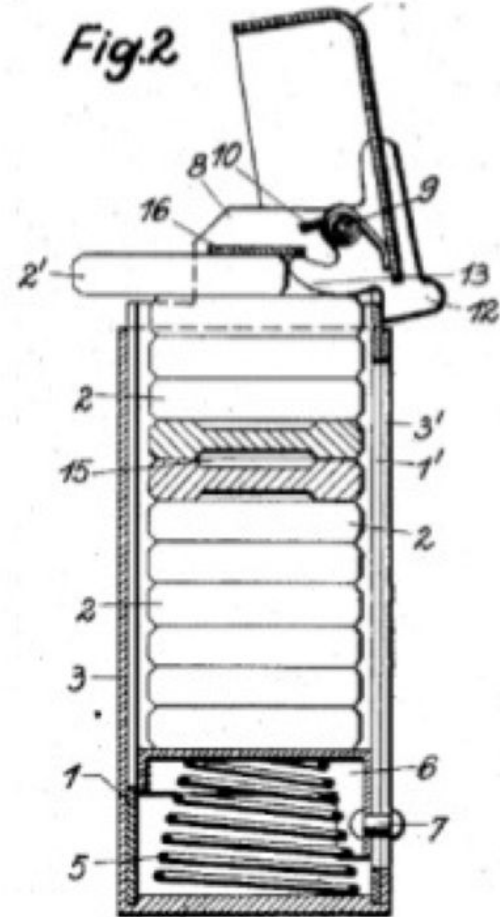
A new ADT: Stack

Programmatic manifestation of _____.

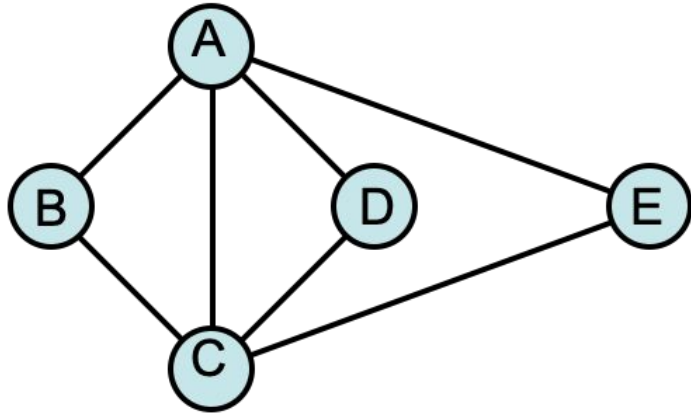
ADT: Stack

Insert -- push(data)

Remove -- pop() returns data



Depth First Search



Algorithm DFS(G,v)

Input: graph G and start vertex v

Output: labeling of the edges of G in the connected component of v as discovery edges and back edges

setLabel(v , VISITED)

For all w in G .adjacentVertices(v)

if getLabel(w) = UNVISITED

 setLabel((v,w) ,DISCOVERY)

 DFS(G,w)

else if getLabel((v,w)) = UNEXPLORED

 setLabel(e ,BACK)

Recursion: An abstract Stack

Moving toward implementation:

Need to be able to check whether a candidate entry is valid.

Suppose we have a variable `grid`, representing the board, and we want to place a value called `num`, in position (x, y) .

Row check:

Column check:

2			
			3
	4	1	

Moving toward implementation:

2			
			3
	4	1	

Need to be able to check whether a candidate entry is valid.

Suppose we have a variable `grid`, representing the board, and we want to place a value called `num`, in position (x, y) .

Region check?

EX: to query a region in a 2d numpy matrix, just define the bounds on the region and use `in`. In the above example, `2 in grid[0:2, 0:2]` returns `True`.

New problem: define the region for given point (x, y) ?

POTD #31 Tue

<https://github.students.cs.ubc.ca/cpsc203-2019w-t1/potd31>

Describe any snags you run into:

1. Line ___: _____
2. Line ___: _____
3. Line ___: _____
4. Line ___: _____
5. Line ___: _____

ToDo for next class...

POTD: Continue every weekday! Submit to repo.

Reading: TLACS Ch 10 & 12 (lists and dictionaries)

References:

<https://brilliant.org/wiki/markov-chains/>

<https://medium.com/@eightlimbed/counting-on-pythons-defaultdict-b652204780bd>