Programming, Problem Solving, and Algorithms

CPSC203, 2019 W1

Announcements

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

* But Proj 3 will likely come out on Fri.

"Problem of the Day" continues!

Today:

Sudoku

~~Markov Chains Fin~~

State Space Search

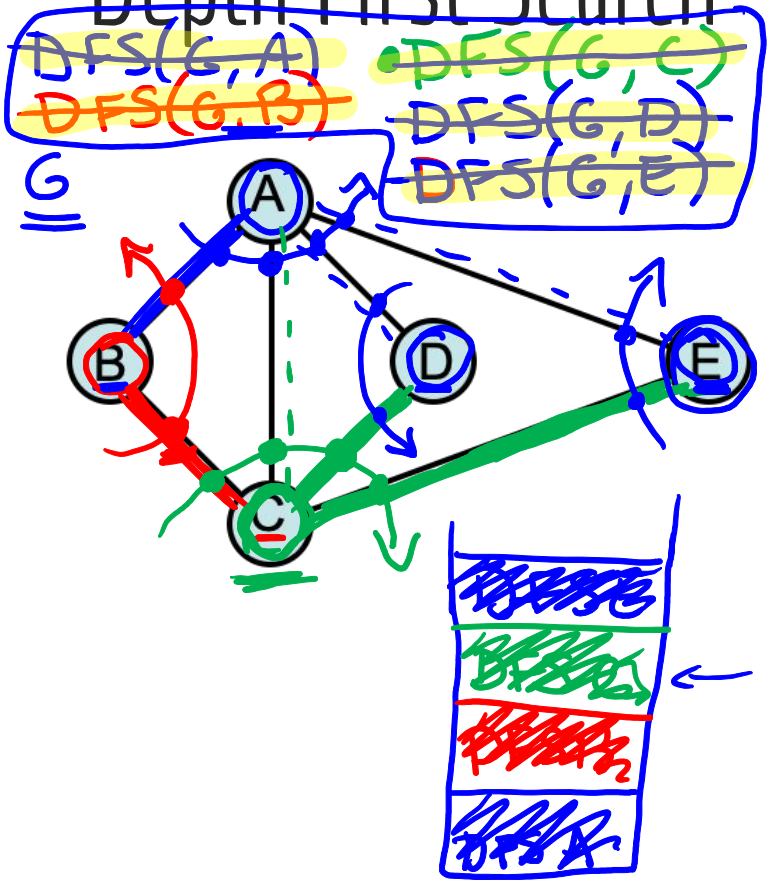
ADT Stack (alternative to a Queue)

~~Representation~~

DFS- Depth First Search

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 a Pez disp.

top

ADT: Stack

Insert -- push(data) places data at top

Remove -- pop() returns data

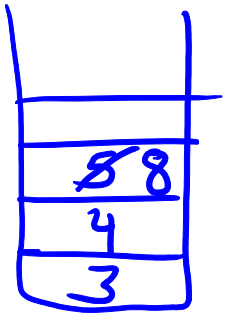
push(3) ✓

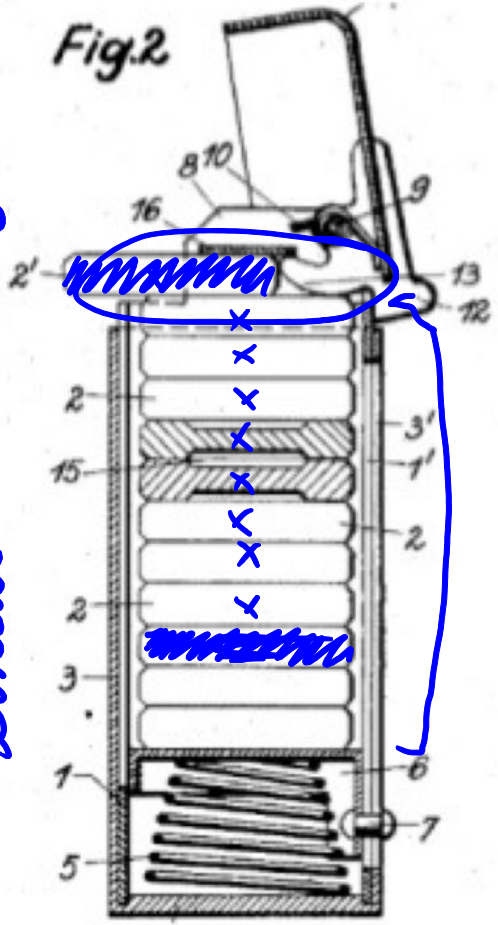
push(4)

push(5) ✓

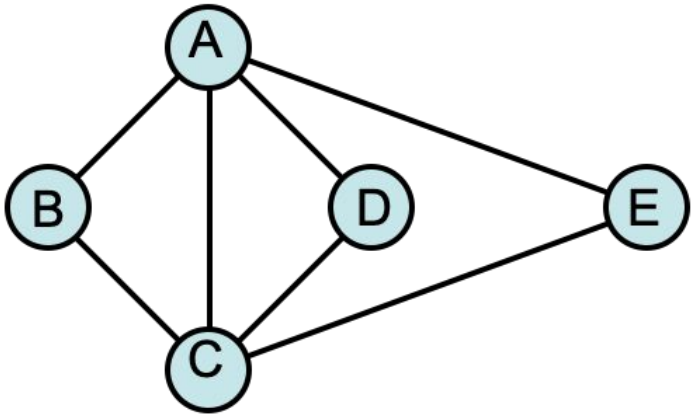
pop() ✓ → 5

push(8) –





Depth First Search



BFS w. Stack instead of a Queue is DFS.

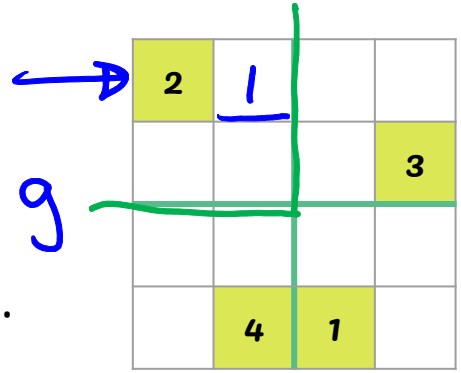
```
Algorithm DFS(G,v)
        Input: graph G and start vertex v
        Output:  labeling of the edges of G in the
connected component of v as discovery edges and
back edges
Stack S;

setLabel(v, VISITED)                    S.push(v)
While !S.isEmpty: x=S.pop()
For all w in G.adjacentVertices(x)
    if getLabel(w) = UNVISITED
        setLabel((x,w),DISCOVERY)
        DFS(G,w)                        S.push(w)
    else if getLabel((x,w)) = UNEXPLORED
        setLabel(e,BACK)
```

Recursion: An abstract Stack

see 1st DFS slide w. stack drawing.

Moving toward implementation:



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

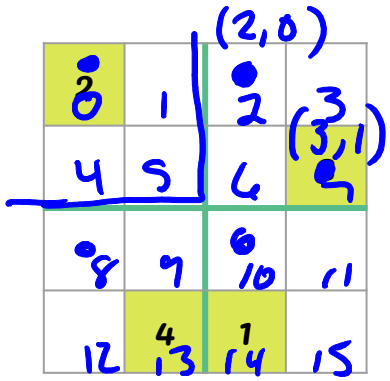
Suppose we have a variable `grid`, representing the board, and we want to place a value called `num`, in position `(x, y)`.

Row check: boolean return num in g[0:, y]

Column check: or g[y, 0:]

bool return num in g[x, 0:]

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)`.

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`. if grid[0,0] or grid[0,1] or ... grid[1,1] is = 2.

→ one past what you want.

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

x//d is integer division. 532//10 → 53

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