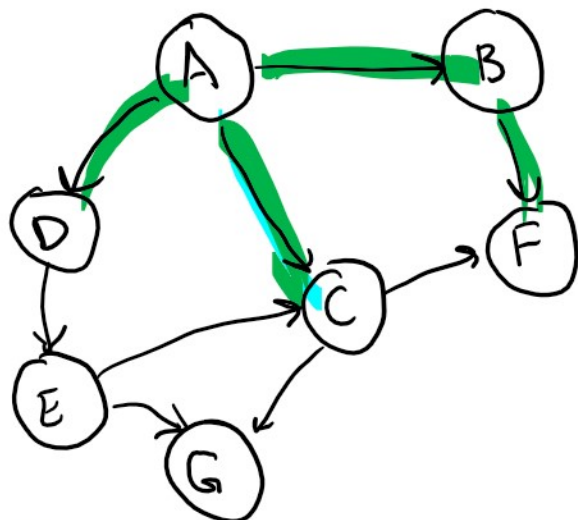


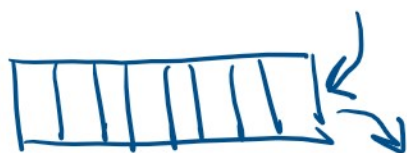
# BFS/DFS example

Wednesday, October 19, 2022 1:18 PM

A → F??



DFS:

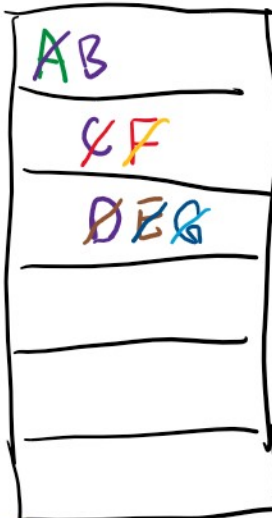


Removing the most recent item to have been added  
"LIFO" (last in/first out)  
stack

Visited:

A, B, C, D, E, G, F

To Check:

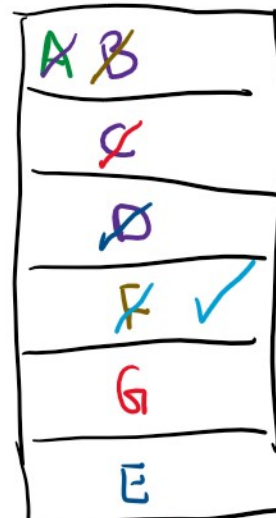


DFS  
(depth-first search)

Visited:

A, B, C, D, F, G, E

To Check:



BFS  
(breadth-first search)

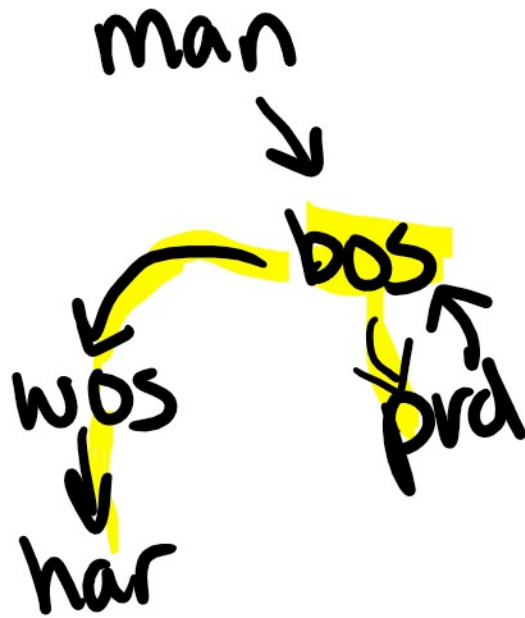
BFS:



Removing the least recent item to have been added  
"FIFO" (first in/first out)  
queue

cameFrom (backtracking on routes)

Friday, October 21, 2022



bos → har

HashMap<CityVertex, CityVertex> cameFrom

prd → bos

wos → bos

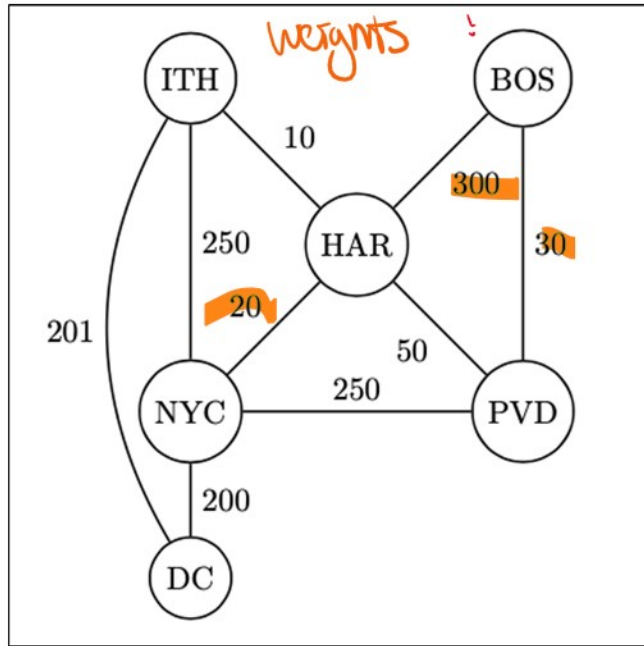
har → wos

bos → wos → har

# Dijkstra

Friday, October 21, 2022 1:06 PM

cheapest way BOS → NYC



"undirected"

to Check PQ

name	rate	distance
<del>BOS</del>	0	
<del>DC</del>	<del>291</del>	
<del>HAR</del>	<del>300</del>	80
<del>ITH</del>	<del>90</del>	
<del>NYC</del>	<del>280</del>	100
<del>PVD</del>	<del>30</del>	

came From

pvd → bos  
 har → bos pvd  
 nyc → pvd har  
 ith → har  
 dc → ith

bos → pvd → har → nyc

Dijkstra's algorithm summary:

Choose a source node (Bos), and at every step, keep track of the **best distance you've seen so far** from the source node to every other node

For every step, consider the cheapest place you can get to, and update the costs to the source node based on the neighbor costs