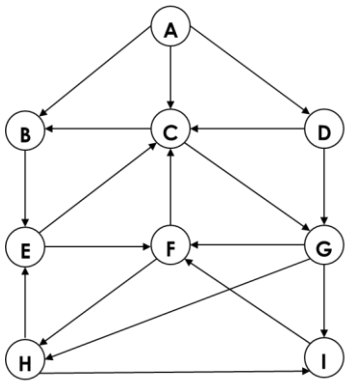
CS3353: Data Structures and Algorithm Analysis I
Spring 2024

Homework #6 – Reference #2

1. Assume that the graph G represents the daily flights between different cities, and we want to fly from city A to I with minimum stops. Find the minimum path P from A to I using a **queue** given that every edge has a length of one. Show all your work.



Adjacency Lists			
A:	B	C	D
B:	E		
C:	B	G	
D:	C	G	
E:	C	F	
F:	C	H	
G:	F	H	I
H:	E	I	
I:	F		

- Use two arrays: QUEUE and ORIG. While QUEUE is used to hold the nodes that have to be processed, ORIG is used to keep track of the origin of each edge.
- Add A to QUEUE and add NULL to ORIG.
  - FRONT = 0; QUEUE = A
  - REAR = 0; ORIG = \0
- Dequeue a node by setting FRONT++ and enqueue the neighbor of A. Also add A as the ORIG of its neighbor.
  - FRONT = 1; QUEUE = A B C D
  - REAR = 3; ORIG = \0 A A A
- Dequeue a node by setting FRONT++ and enqueue the neighbor of B. Also add B as the ORIG of its neighbor.
  - FRONT = 2; QUEUE = A B C D E
  - REAR = 4; ORIG = \0 A A A B
- Dequeue a node by setting FRONT++ and enqueue the neighbor of C. Also add C as the ORIG of its neighbor.
  - FRONT = 3; QUEUE = A B C D E G
  - REAR = 5; ORIG = \0 A A A B C
- Dequeue a node by setting FRONT++ and enqueue the neighbor of D. Also add D as the ORIG of its neighbor.
  - FRONT = 4; QUEUE = A B C D E G
  - REAR = 5; ORIG = \0 A A A B C
- Dequeue a node by setting FRONT++ and enqueue the neighbor of E. Also add E as the ORIG of its neighbor.
  - FRONT = 5; QUEUE = A B C D E G F
  - REAR = 6; ORIG = \0 A A A B C E
- Dequeue a node by setting FRONT++ and enqueue the neighbor of G. Also add G as the ORIG of its neighbor.
  - FRONT = 1; QUEUE = A B C D E G F H I
  - REAR = 3; ORIG = \0 A A A B C E G G
- As I is our final destination, we stop the execution. Now backtrack from I using ORIG to find the minimum path P,
  - A → C → G → I

