

# Breadth first Traversal

DFT → Recursion

⇓

Stack

Data

Structure

**Queue**

~~A~~ ~~B~~ ~~C~~ ~~D~~ E f

FIFO

Queue

1	1	1	1	1	1	
<del>0</del>	<del>0</del>	<del>0</del>	<del>0</del>	<del>0</del>	<del>0</del>	0
A	B	C	D	E	f	G

x = A

**BFT(A)**

inside  
the while  
loop

for all w adj x:

if w not visited:

visited(w) = 1

Add(w, Q)

A, B, C, D

A - [B, C, D]

B - [A, E]

C - [A, E, f]

D - [A, f]

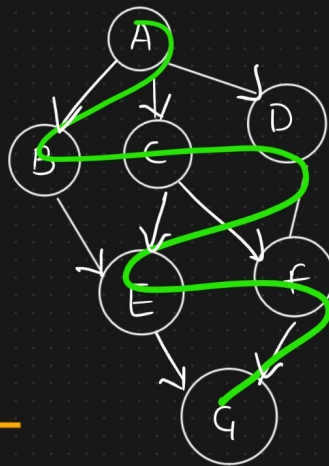
E - [B, C, G]

f - [C, D, G]

G - [E, f]

1) Breadth first Traversal

2) Greedy Algorithms  
→ fractional Knapsack



BFT(v):

visited(v) = 1

Add(v, Q)

while (Q not empty):

x = delete(Q)

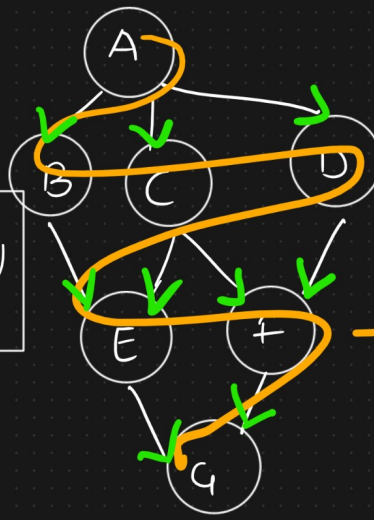
Print(x)

visited  
↳ set

A, B, C, D, E, f, G

Queue (FIFO)

~~A~~ ~~B~~ ~~C~~ ~~D~~ ~~E~~ ~~f~~ ~~G~~



BFT  
↳ Queue

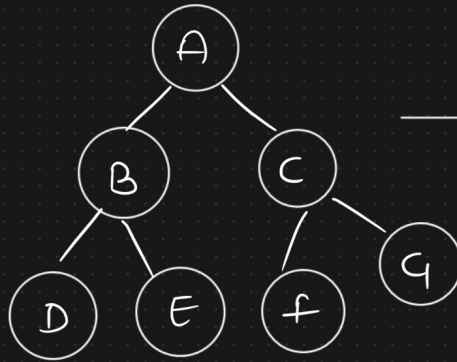
BFT(A)

Level order  
Traversal

A, B, C, D, E, f, G

↳ BFT

Task  
↳



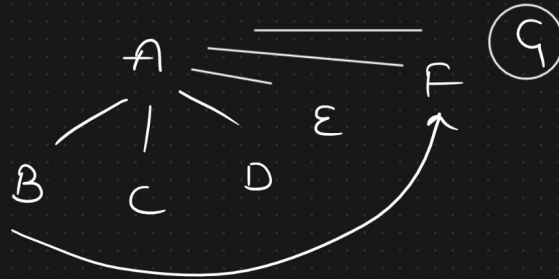
Level order  
Traversal

'A B C D E f G'

## Time complexity

### Adjacency List

$$\underline{\underline{\text{BFT :- } V + 2E}}$$

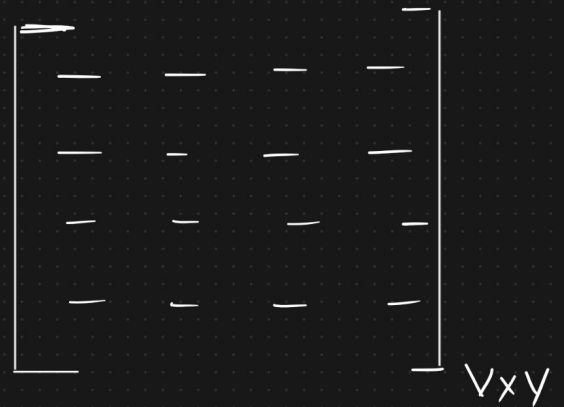


$$\Downarrow$$
$$\underline{\underline{\Theta(V+E)}}$$

### Adjacency Matrix

$$V + V^2$$

$$\underline{\underline{\Theta(V^2)}}$$



### DFT

$$\underline{\underline{\text{Adjacency List}}} - V + 2E$$

$$\Theta(V+E)$$

$$\underline{\underline{\text{Adjacency Matrix}}} - V + V^2$$

$$\Theta(V^2)$$

Sparse Matrix

↳ Most of elements are 0