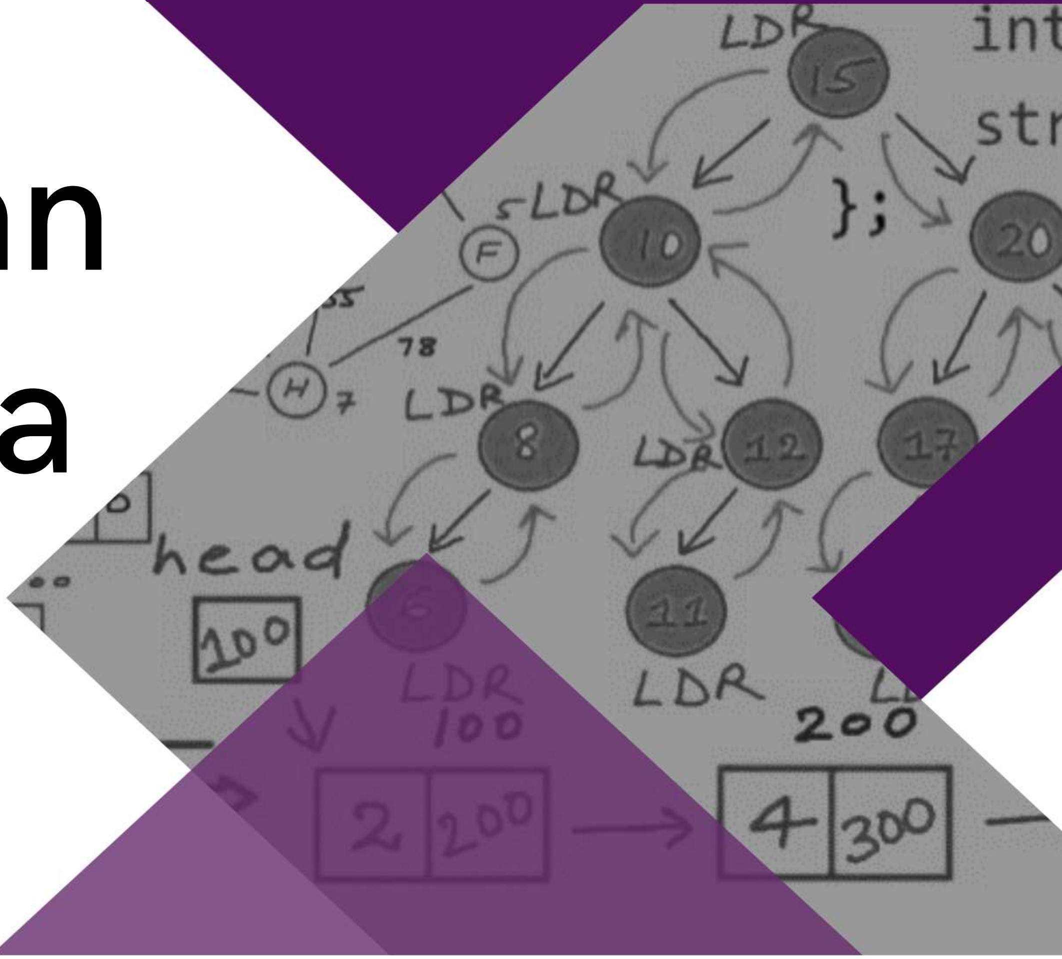
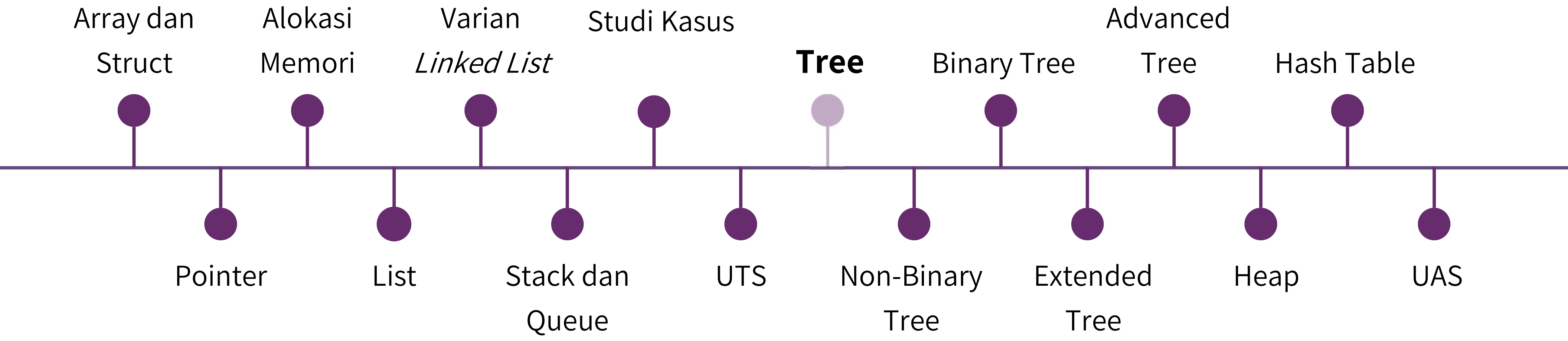
Algoritma dan Struktur Data



Pekan 9



Array dan Struct

Pointer

Alokasi Memori

List

Varian *Linked List*

Stack dan Queue

Studi Kasus

UTS

Tree

Non-Binary Tree

Binary Tree

Extended Tree

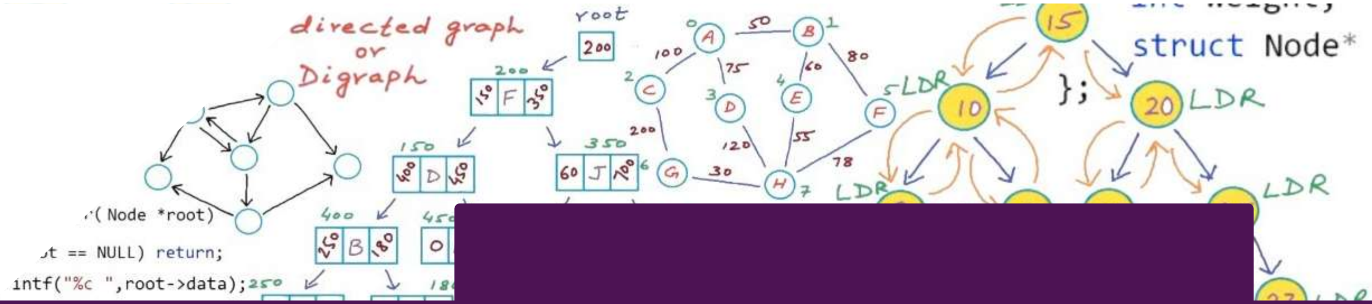
Advanced Tree

Heap

Hash Table

UAS

Tujuan

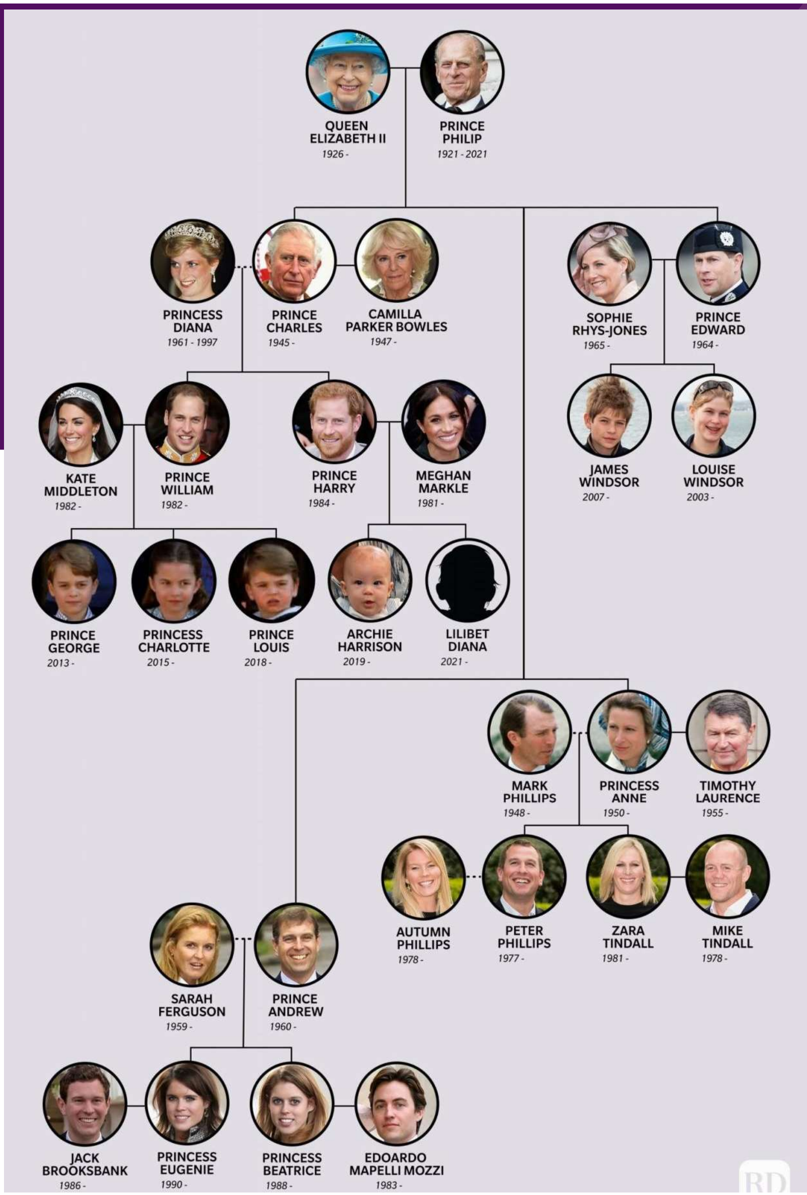


1	Mahasiswa memahami perbedaan struktur linear dan hirarki
2	Mahasiswa memahami permasalahan yang membutuhkan struktur hirarki
3	Mahasiswa mengetahui terminology pada struktur data hirarki
4	Mahasiswa memahami proses dasar pada struktur hirarki
5	Mahasiswa mampu menerapkan struktur hirarki pada kasus nyata

Penggunaan Tree

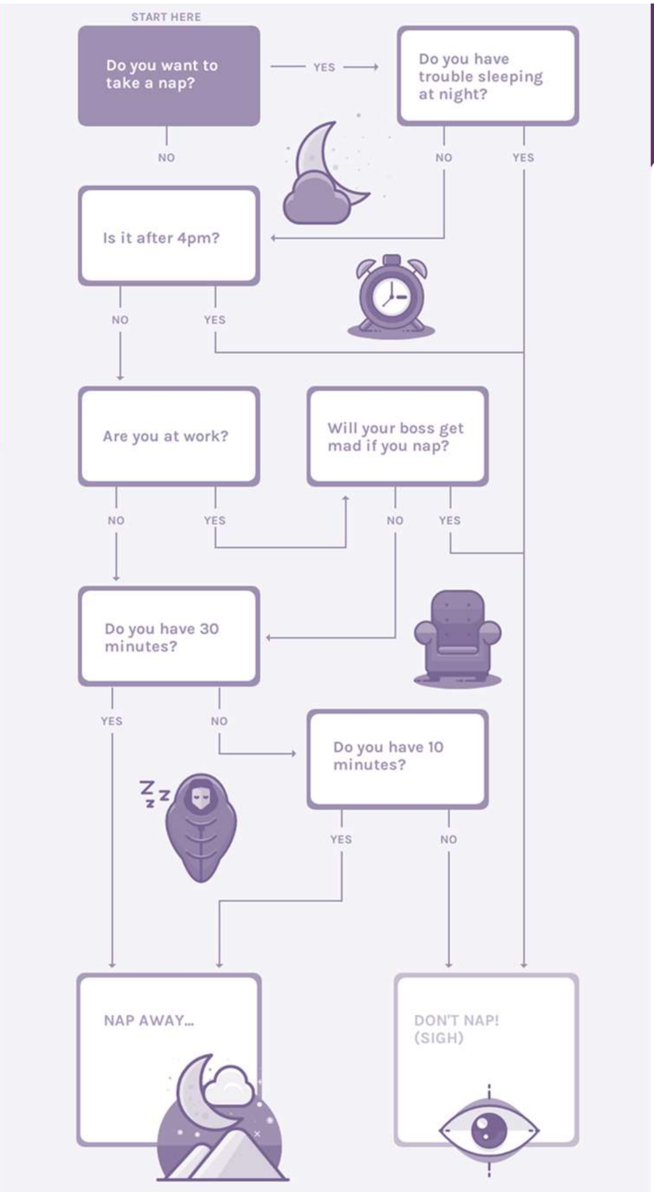
Penggunaan Tree

Family Tree



Penggunaan Tree

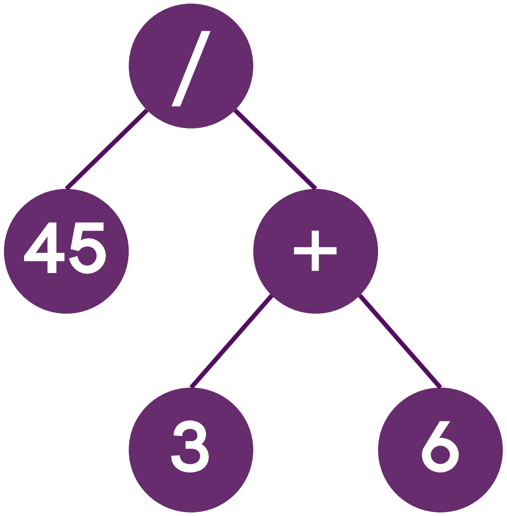
Decision Tree



Penggunaan Tree

Expression Tree

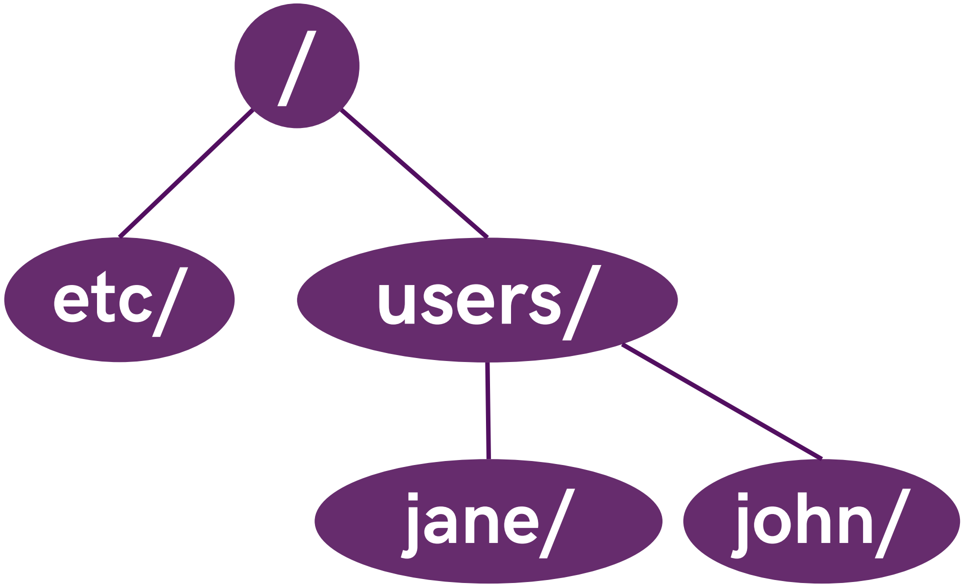
45 / (3 + 6)



Penggunaan Tree

File System

/users/jane/



Trees

Tons of trees in computer science

Regular trees

Binary trees

Heap

Binary Search Trees

Huffman Trees

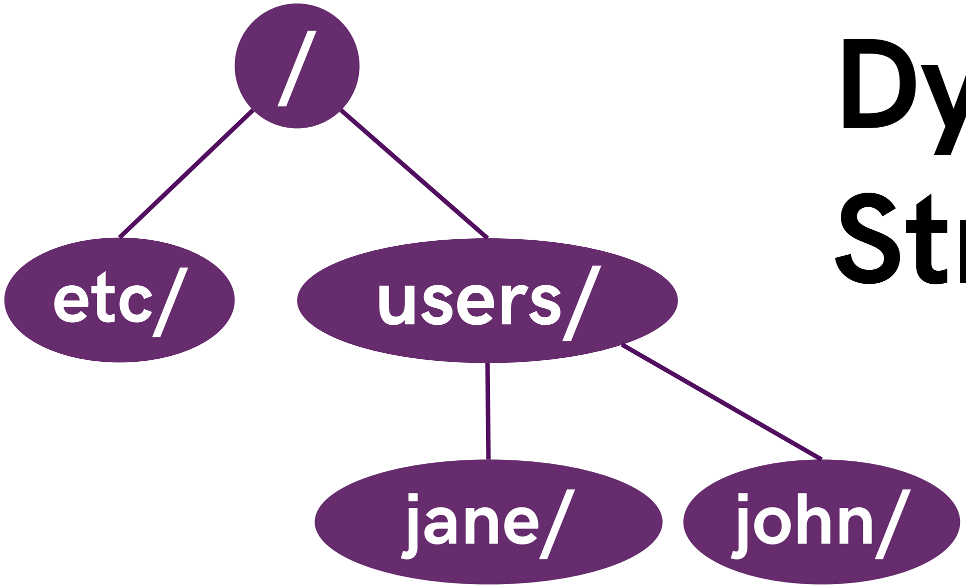
AVL Trees

Randomized Search Trees

Red-Black Trees

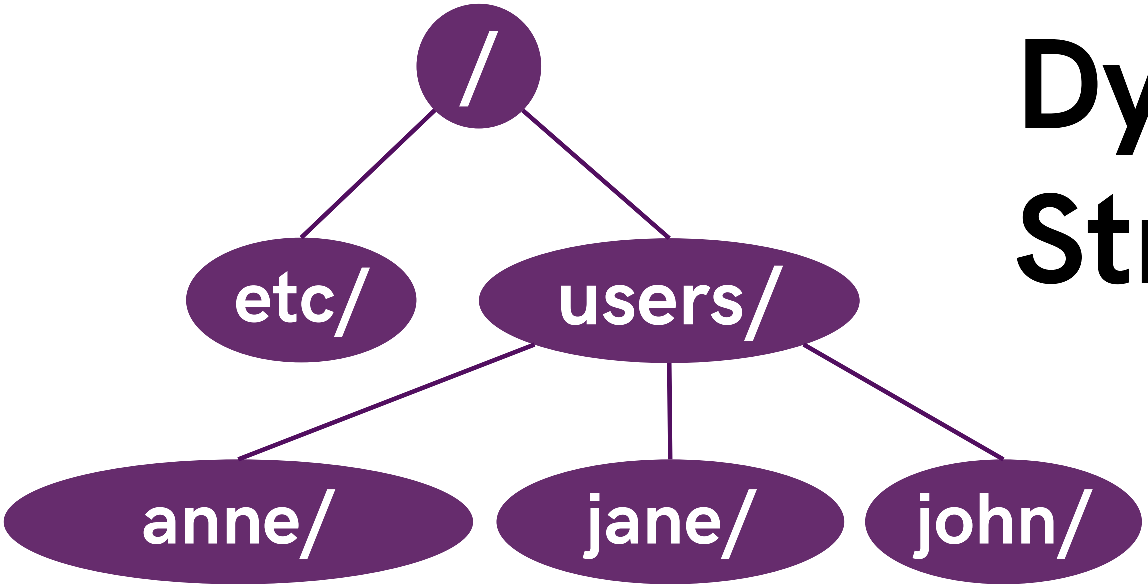
Karakteristik Tree

Why Trees?



Dynamic Data Structure

Why Trees?



Dynamic Data Structure

Why Trees?

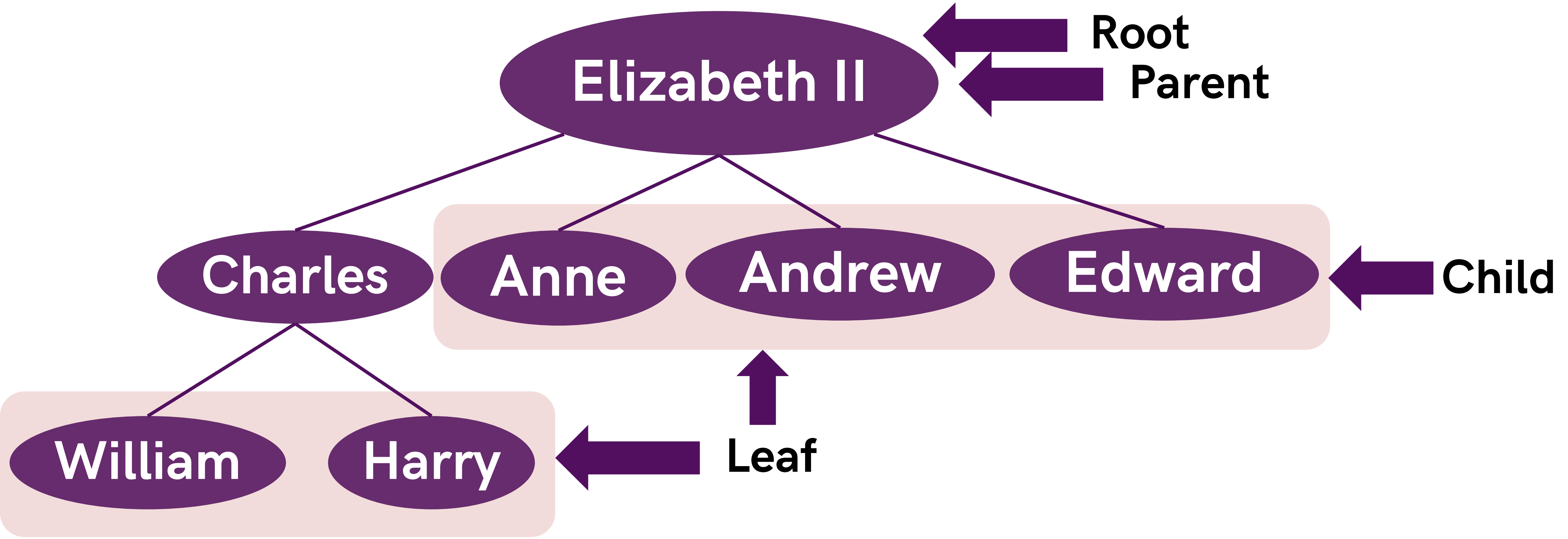
Structure Convey Information

Different organization → Different Trees

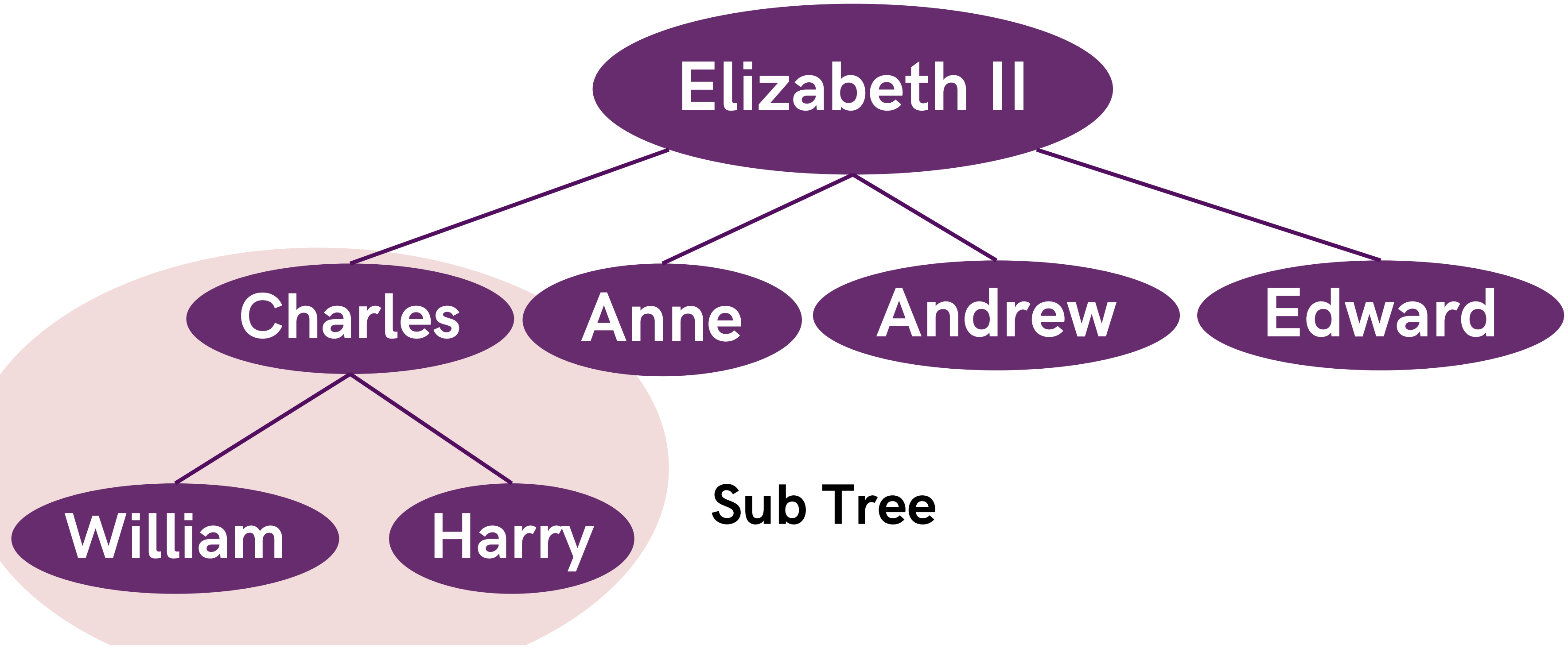
- Root is most important → **Heap**
- Organized by character frequency → **Huffman Trees**
- Organized by node ordering → **Search Trees**
- Etc…

Terminologi

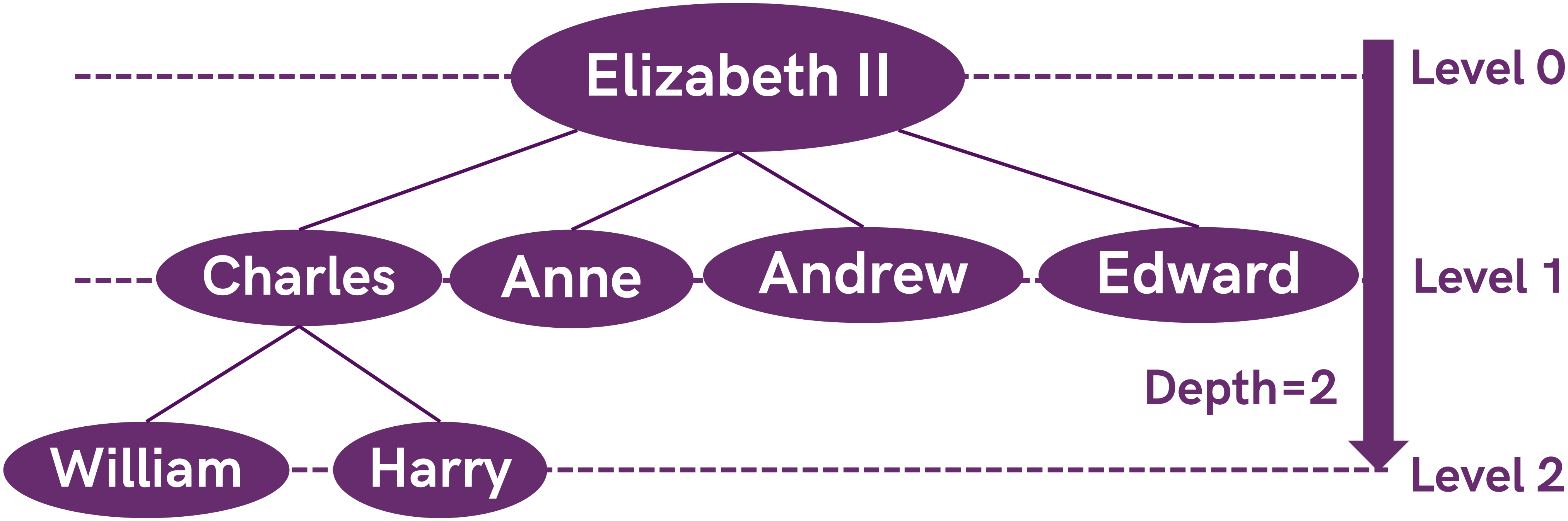
Terminologi



Terminologi



Terminologi



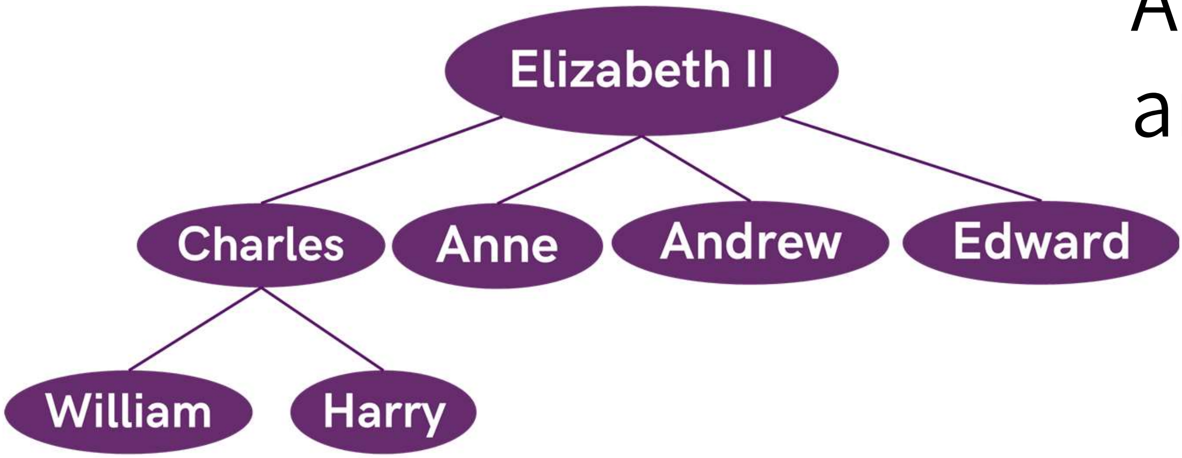
Istilah	Definisi
Node	Sebuah elemen dalam sebuah tree; berisi sebuah informasi.
Parent	Node yang berada di atas node lain secara langsung
Child	Cabang langsung dari sebuah node.
Sibling	Sebuah node lain yang berada pada level yang sama dan memiliki parent yang sama
Root	Node paling atas pada tree; sebuah node yang tidak memiliki parent
Leaf	Sebuah node pada tree yang tidak memiliki children
Level	Jarak dari sebuah node ke root; root berada pada level 0
Depth	Maksimum level pada sebuah tree

What Defines a Tree?

- Single Root
- Each node can have **only one** parent
- No cycles in a tree

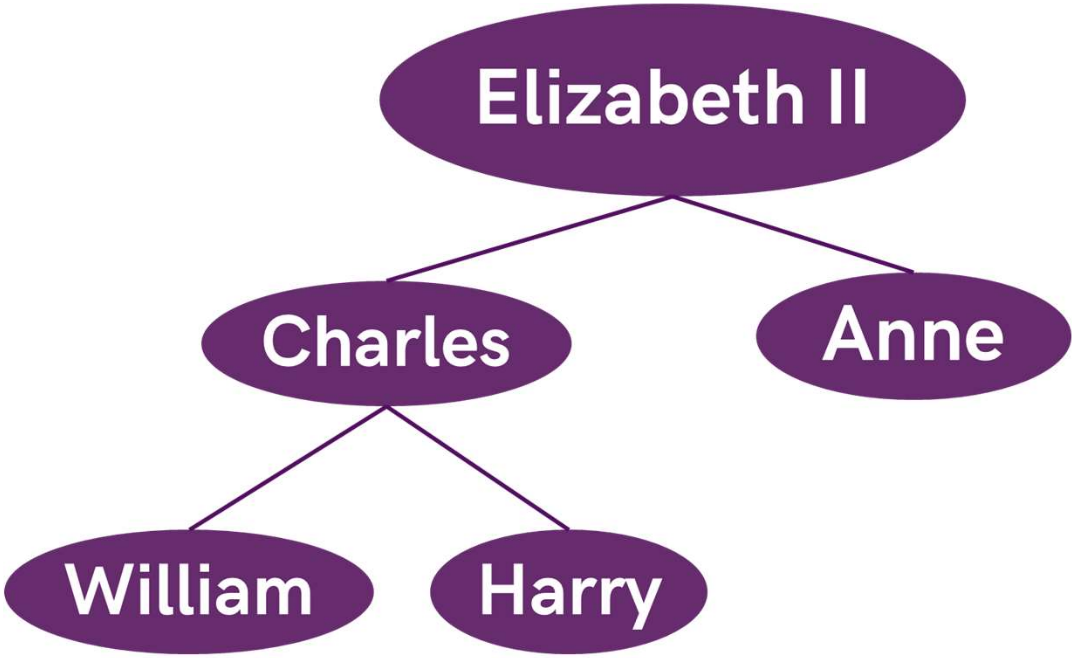
Jenis Tree

Generic Tree



Any parent can have any number of children

Binary Tree



Any parent can have **at most** two children

Tree Traversal

Traversal

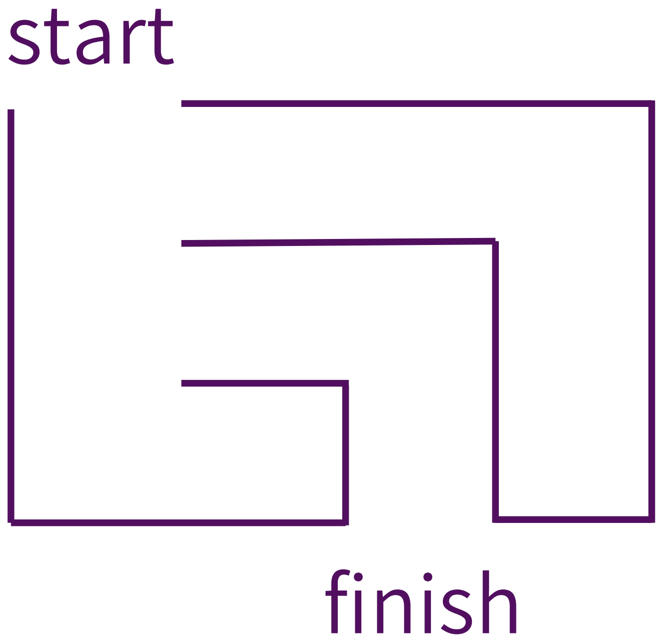
Berdasarkan cara: (digunakan untuk graph)

- Breadth First Traversal
- Depth First Traversal

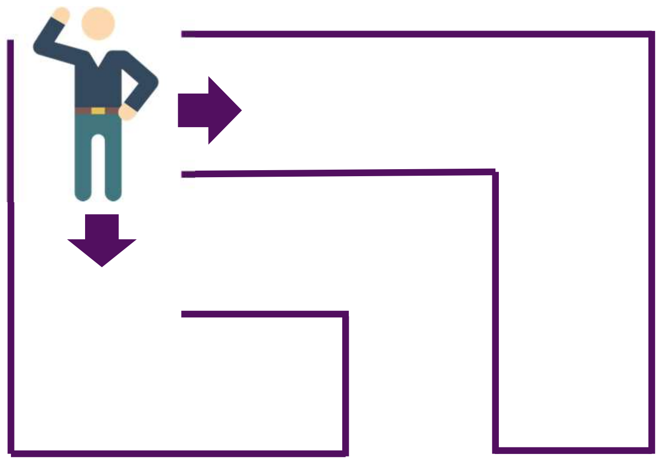
Berdasarkan urutan: (digunakan untuk tree)

- Pre-order Traversal
- Post-order Traversal
- In-order Traversal
- Level-order Traversal

Contoh - Labirin



Contoh - Labirin

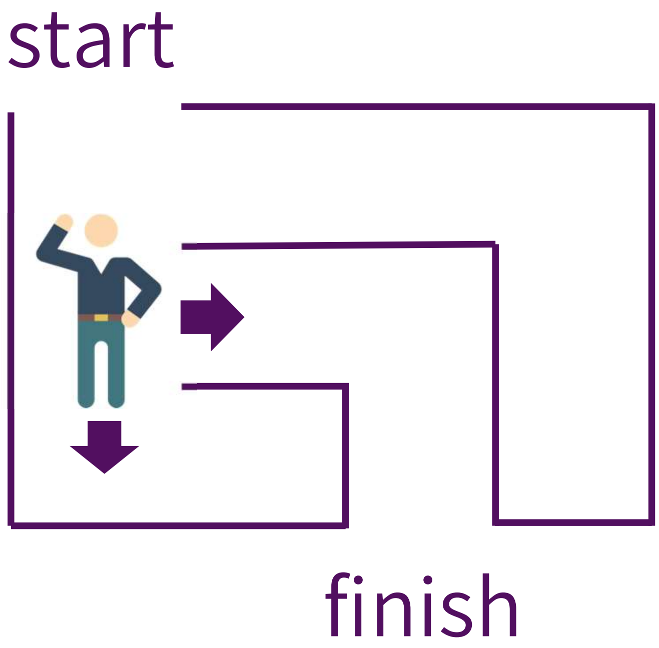


start

finish

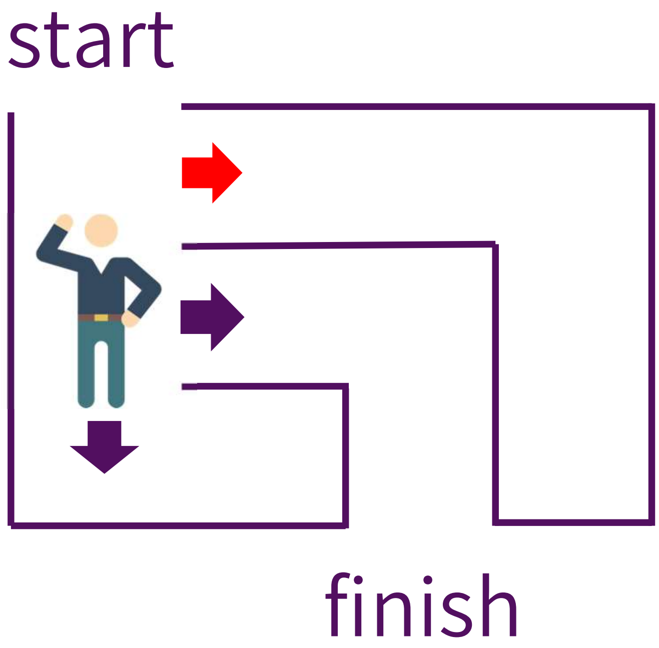
What’s my next step?

Contoh - Labirin



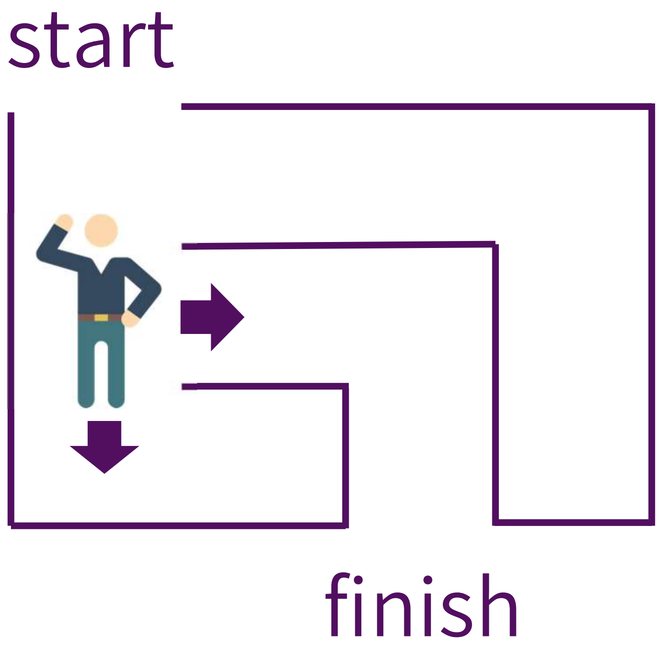
What’s my next step?

Contoh - Labirin



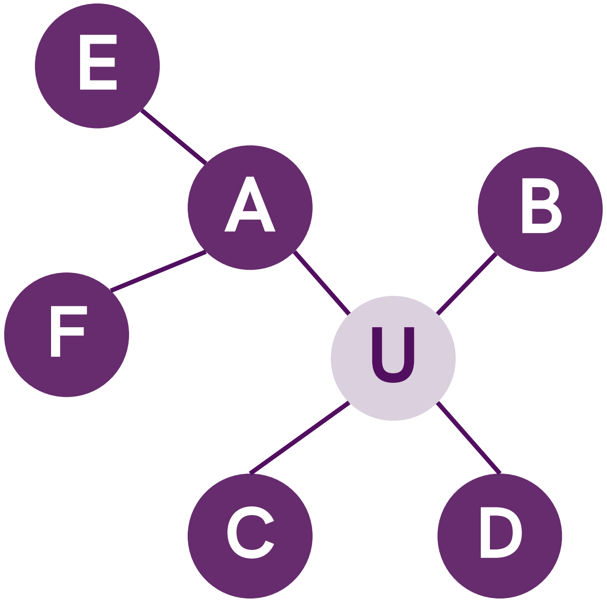
What’s my next step?

Contoh - Labirin



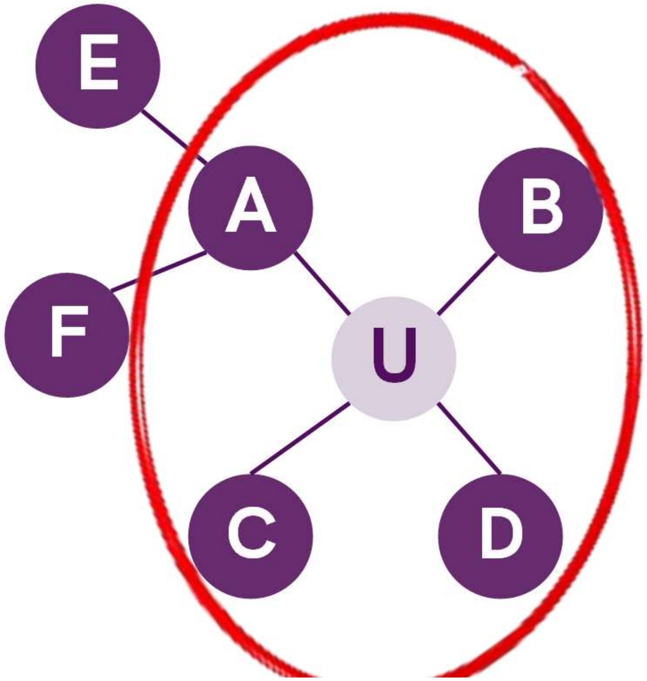
Labirin menggunakan penelusuran **Depth First Traversal**.

Contoh – Jaringan Sosial



Seberapa dekat anda terhubung dengan D?

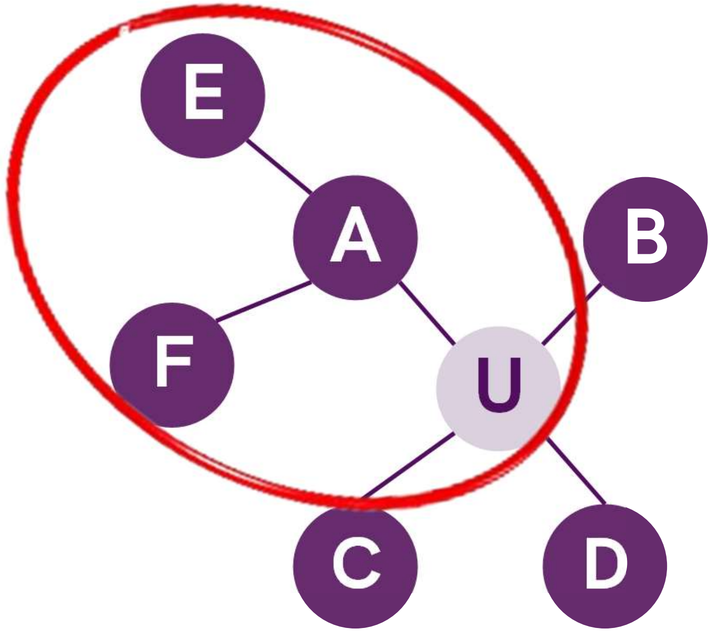
Contoh – Jaringan Sosial



Seberapa dekat anda terhubung dengan D?

Misalkan anda memiliki daftar teman-teman dan setiap teman anda memiliki daftar teman-teman lainnya.

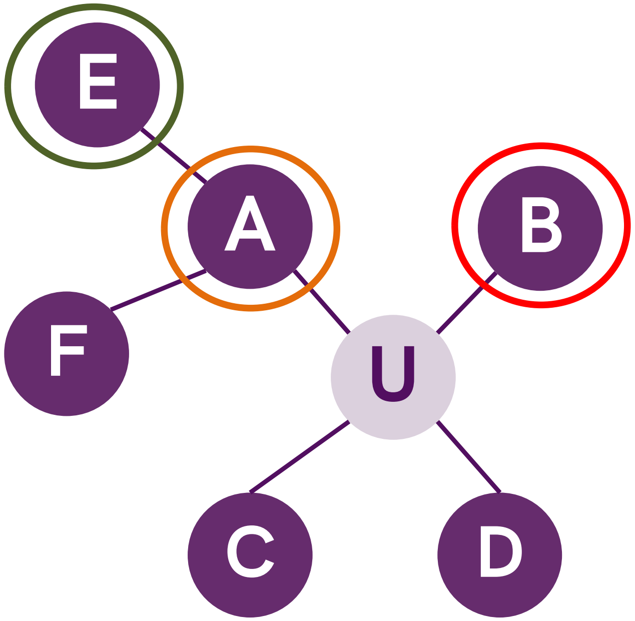
Contoh – Jaringan Sosial



Seberapa dekat anda terhubung dengan D?

Misalkan anda memiliki daftar teman-teman dan setiap teman anda memiliki daftar teman-teman lainnya.

Contoh – Jaringan Sosial



Seberapa dekat anda terhubung dengan D?

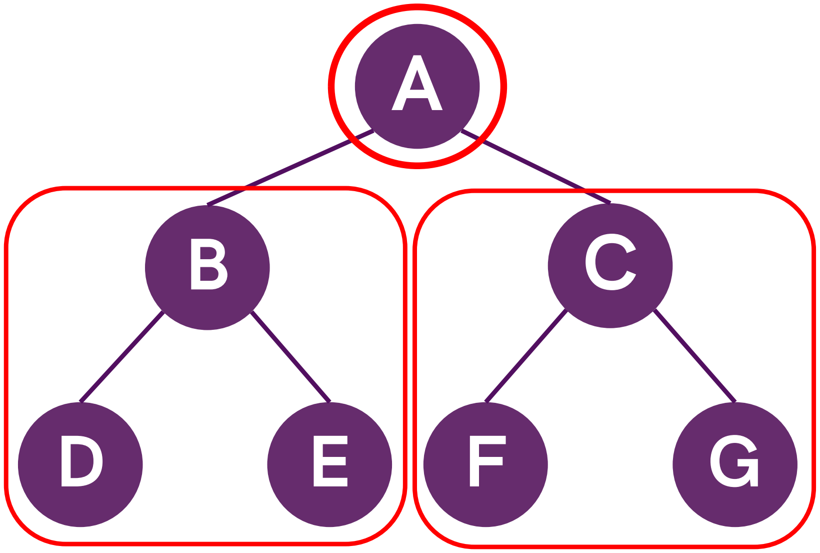
Apa langkah berikutnya?

Look at all of your friends first, and then branch out.

Kesimpulan

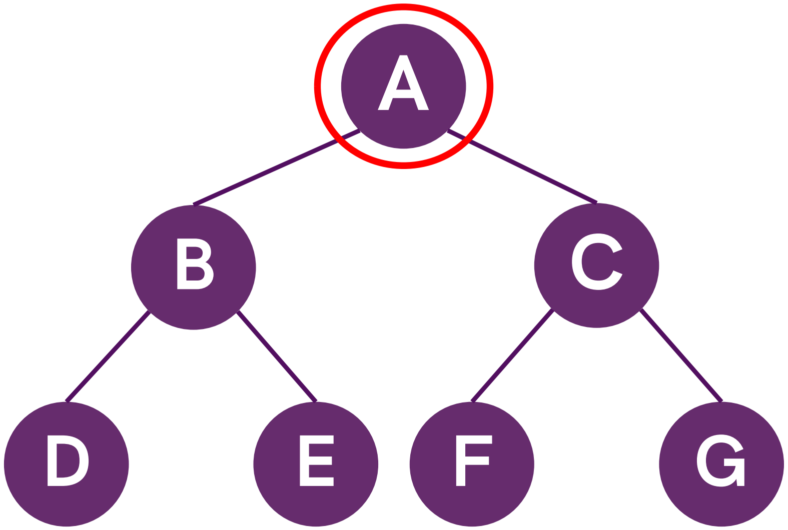
Urutan node yang dikunjungi merupakan hal penting dan keputusan urutan node yang dikunjungi didasarkan pada kebutuhan atau permasalahan yang ingin diselesaikan.

Pre-Order Traversal



- Visit yourself
- Visit all your left subtree
- Visit all your right subtree

Pre-Order Traversal

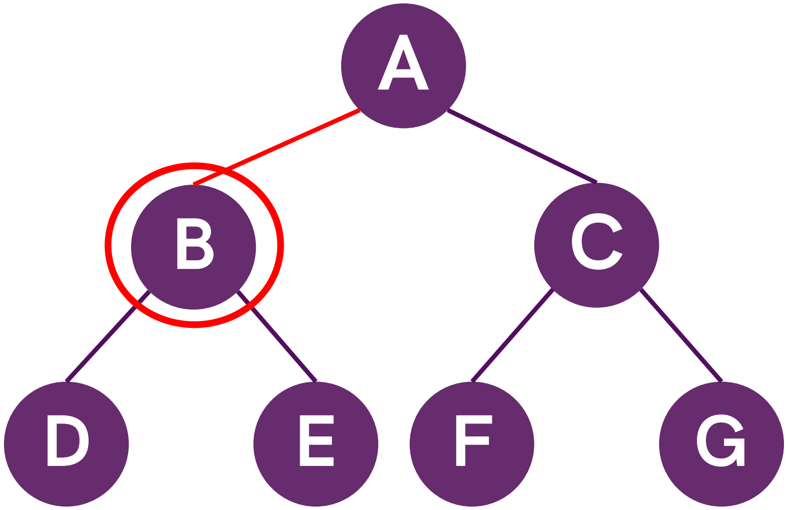


- Visit yourself
- Visit all your left subtree
- Visit all your right subtree

Visited:

A

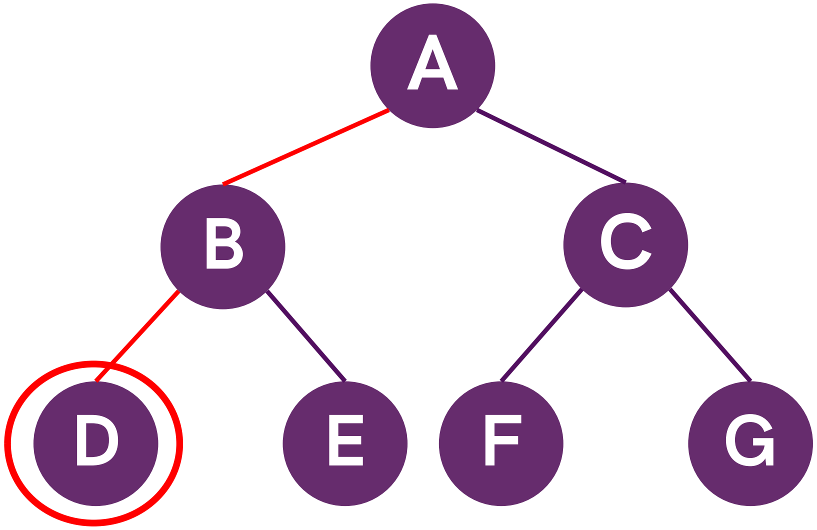
Pre-Order Traversal



- Visit yourself
- Visit all your left subtree
- Visit all your right subtree

Visited:
A B

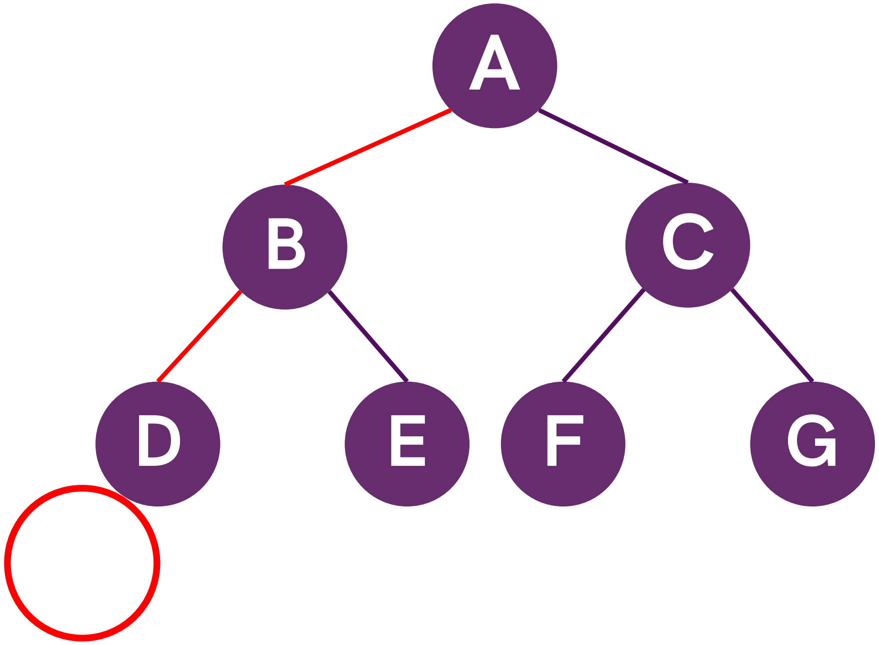
Pre-Order Traversal



- Visit yourself
- Visit all your left subtree
- Visit all your right subtree

Visited:
A B D

Pre-Order Traversal

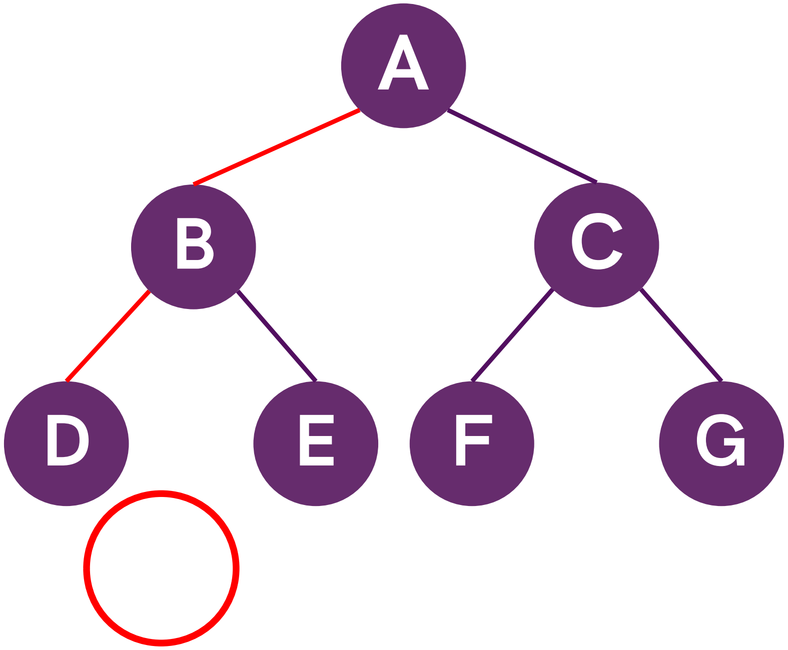


- Visit yourself
- Visit all your left subtree
- Visit all your right subtree

Visited:

A B D

Pre-Order Traversal

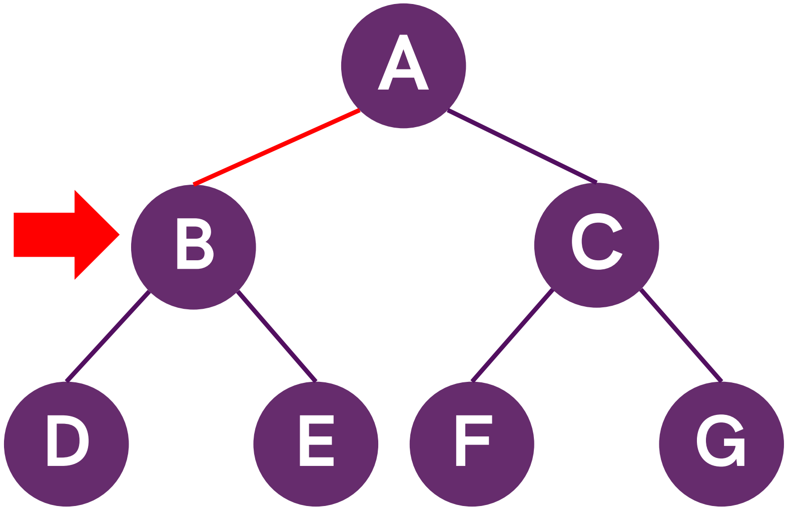


- Visit yourself
- Visit all your left subtree
- Visit all your right subtree

Visited:

A B D

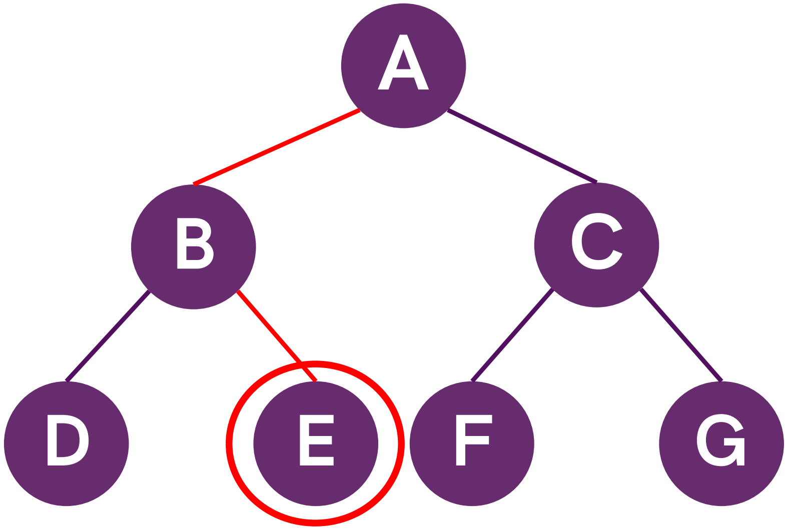
Pre-Order Traversal



- Visit yourself
- Visit all your left subtree
- Visit all your right subtree

Visited:
A B D

Pre-Order Traversal

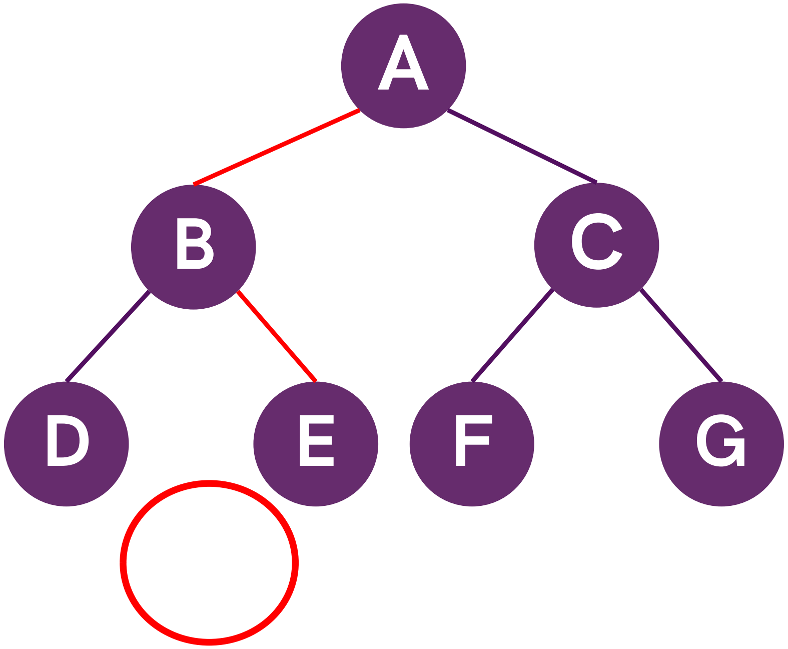


- Visit yourself
- Visit all your left subtree
- Visit all your right subtree

Visited:

A B D E

Pre-Order Traversal

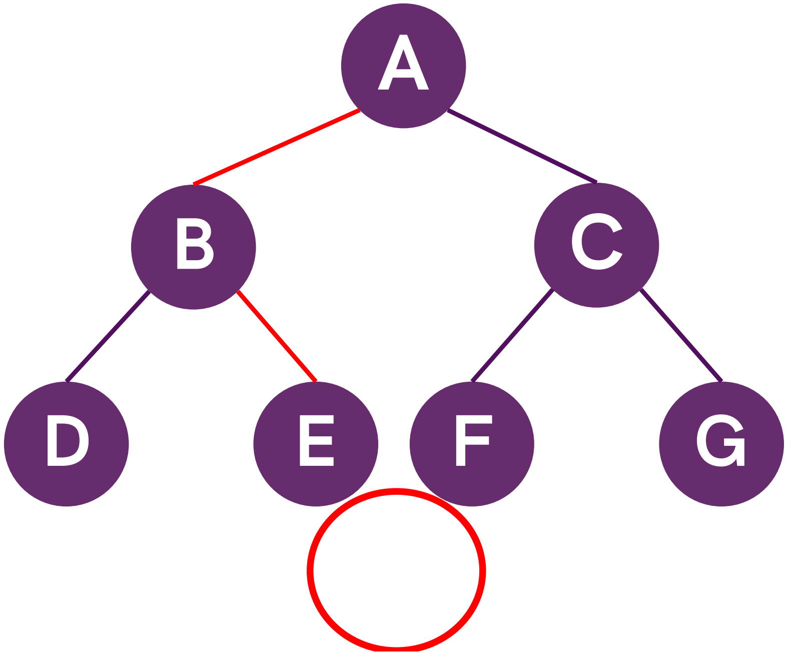


- Visit yourself
- Visit all your left subtree
- Visit all your right subtree

Visited:

A B D E

Pre-Order Traversal

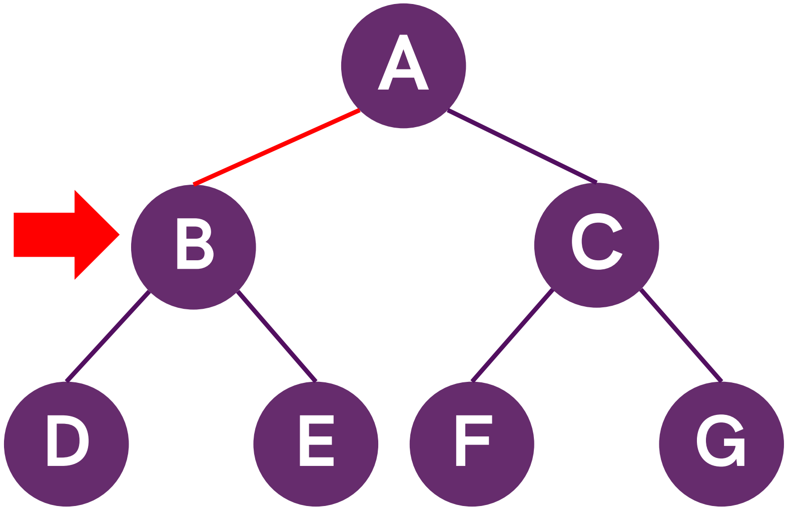


- Visit yourself
- Visit all your left subtree
- Visit all your right subtree

Visited:

A B D E

Pre-Order Traversal

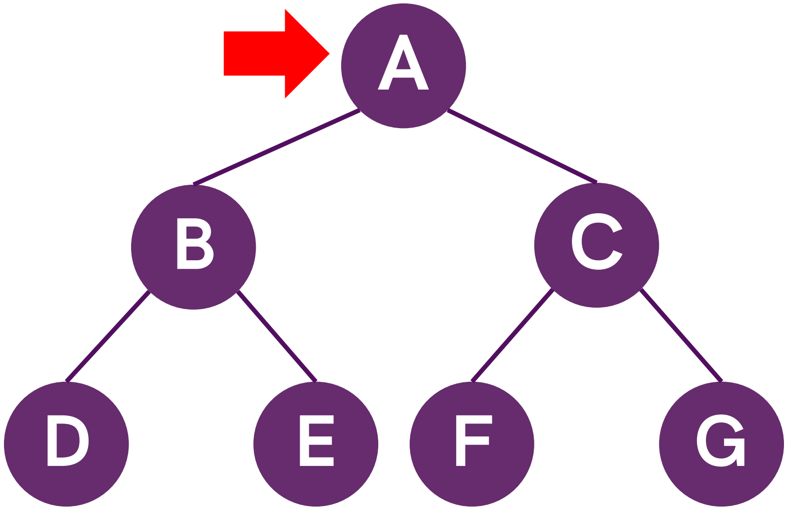


- Visit yourself
- Visit all your left subtree
- Visit all your right subtree

Visited:

A B D E

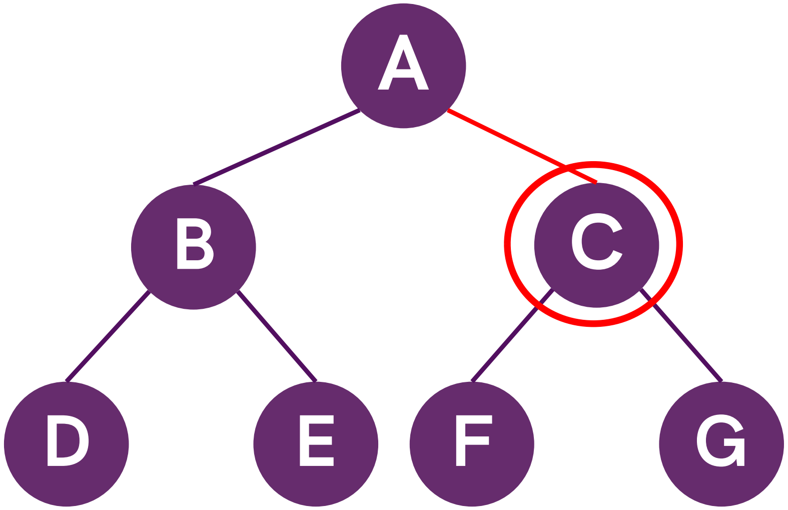
Pre-Order Traversal



- Visit yourself
- Visit all your left subtree
- Visit all your right subtree

Visited:
A B D E

Pre-Order Traversal

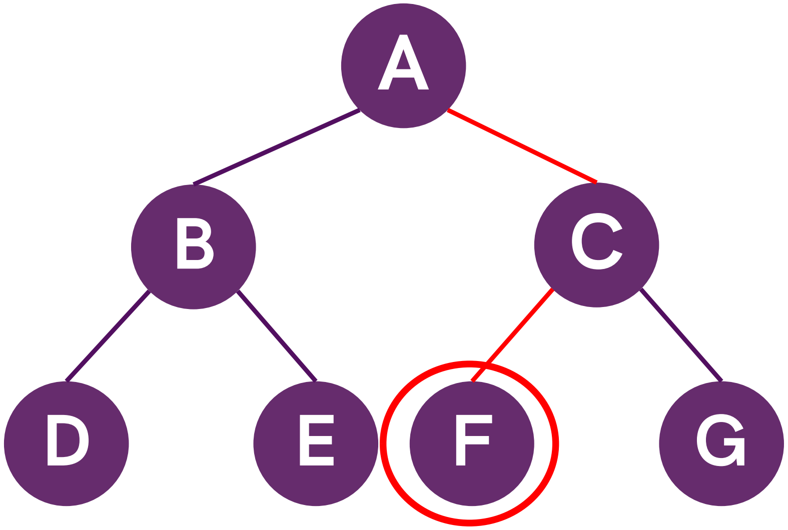


- Visit yourself
- Visit all your left subtree
- Visit all your right subtree

Visited:

A B D E C

Pre-Order Traversal

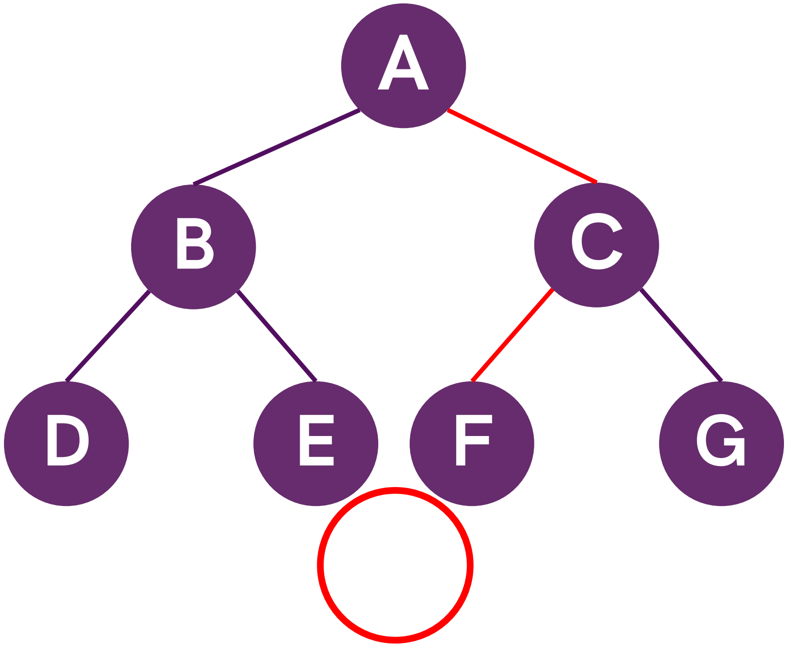


- Visit yourself
- Visit all your left subtree
- Visit all your right subtree

Visited:

A B D E C F

Pre-Order Traversal

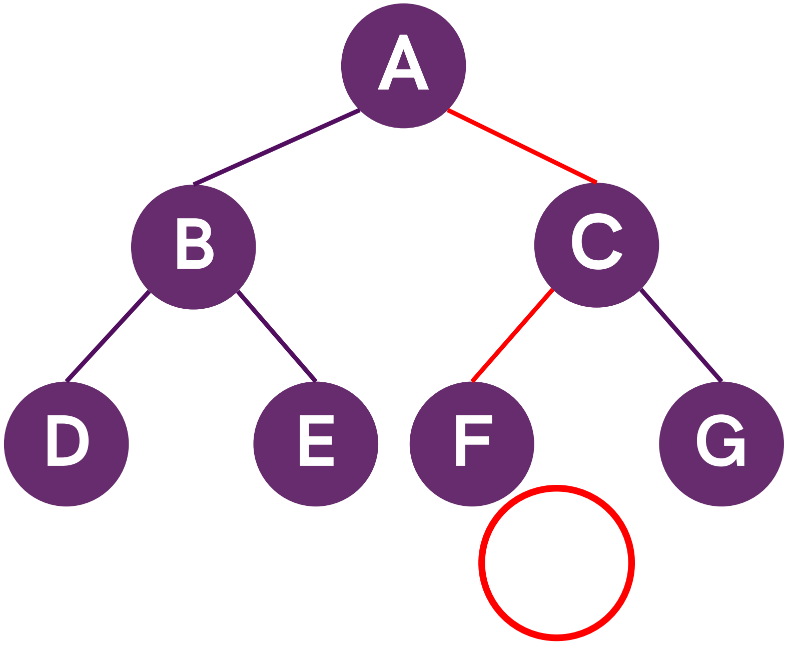


- Visit yourself
- Visit all your left subtree
- Visit all your right subtree

Visited:

A B D E C F

Pre-Order Traversal

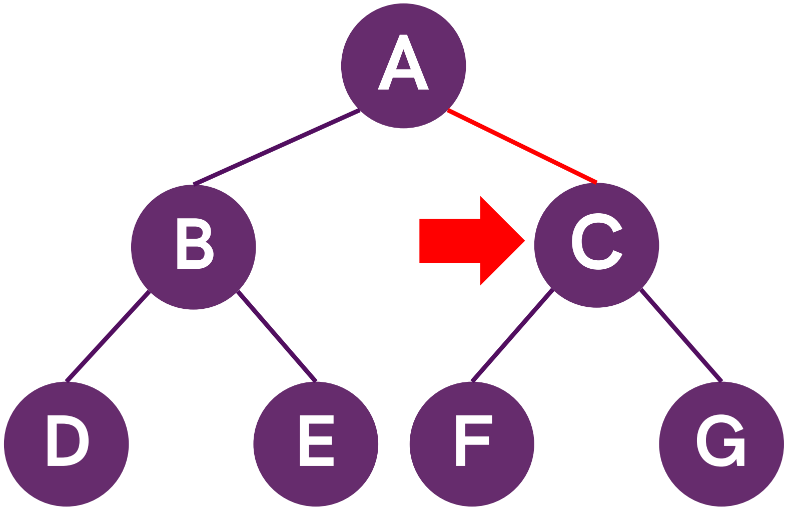


- Visit yourself
- Visit all your left subtree
- Visit all your right subtree

Visited:

A B D E C F

Pre-Order Traversal

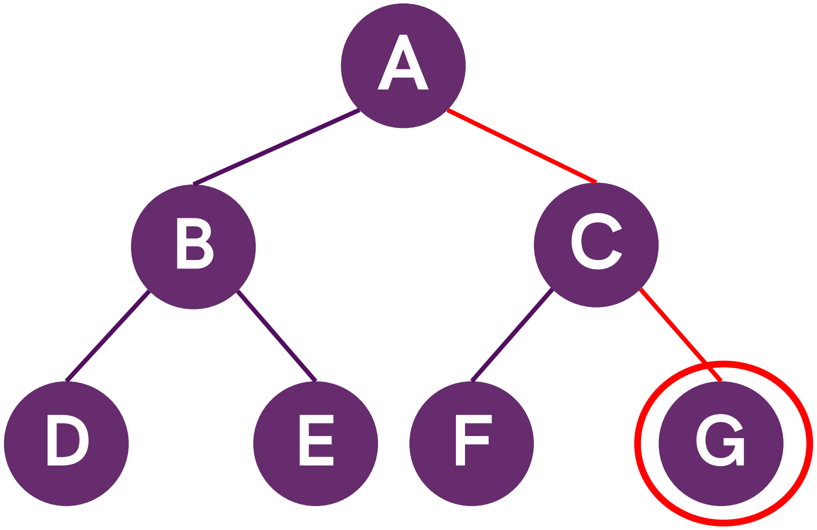


- Visit yourself
- Visit all your left subtree
- Visit all your right subtree

Visited:

A B D E C F

Pre-Order Traversal

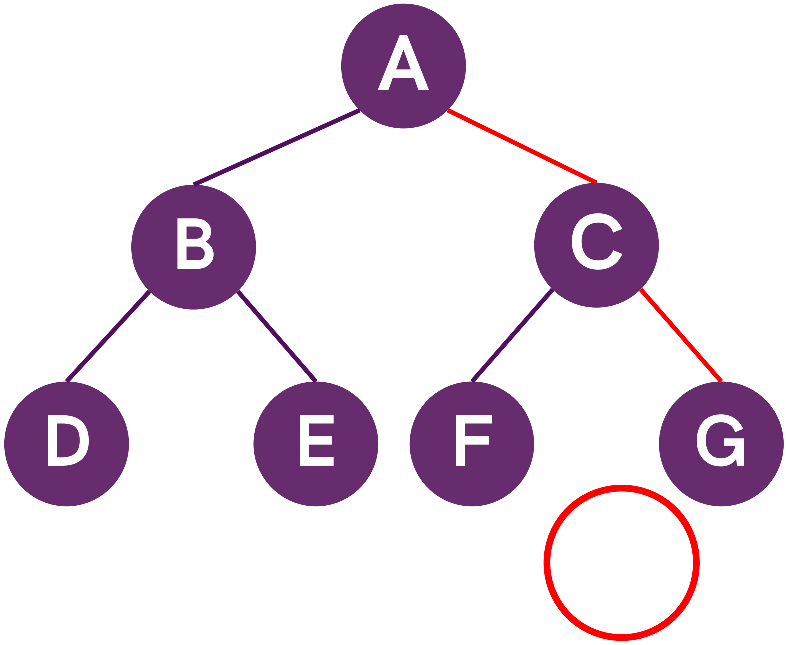


- Visit yourself
- Visit all your left subtree
- Visit all your right subtree

Visited:

A B D E C F G

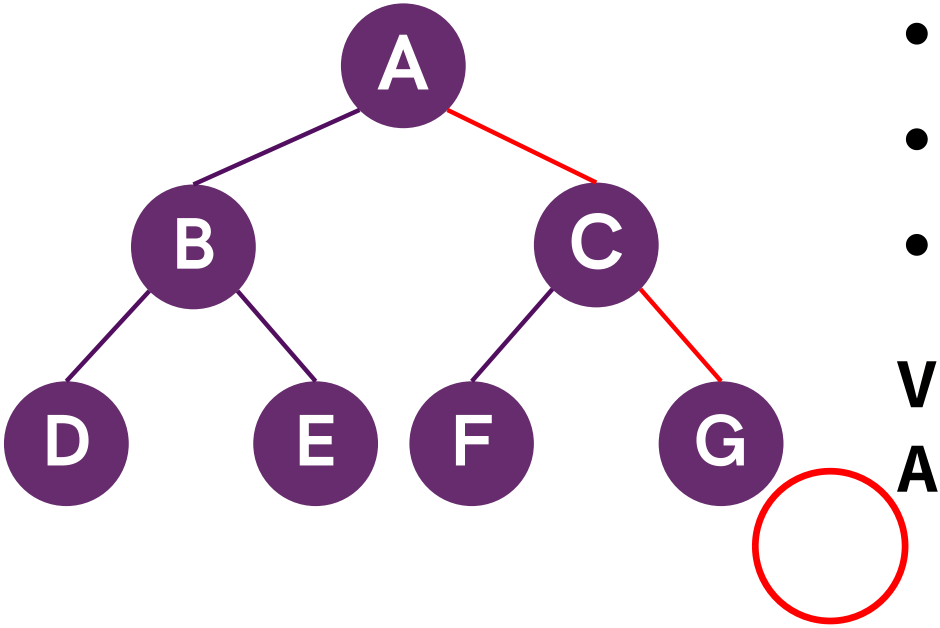
Pre-Order Traversal



- Visit yourself
- Visit all your left subtree
- Visit all your right subtree

Visited:
A B D E C F G

Pre-Order Traversal

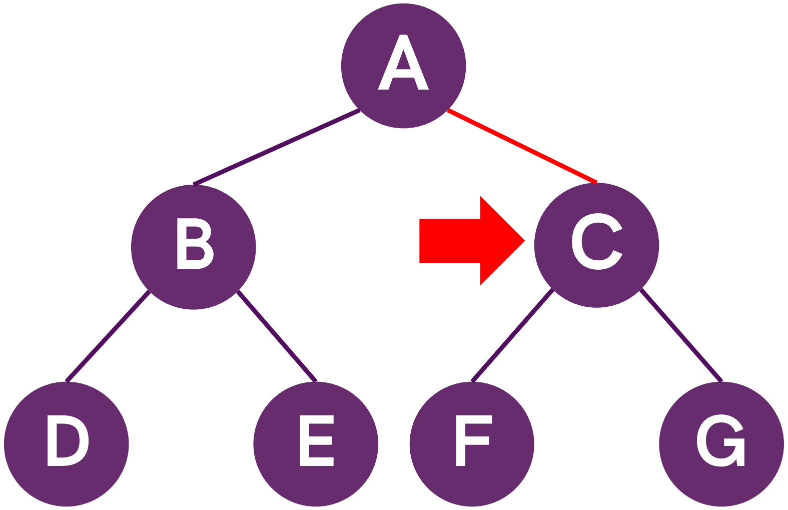


- Visit yourself
- Visit all your left subtree
- Visit all your right subtree

Visited:

A B D E C F G

Pre-Order Traversal

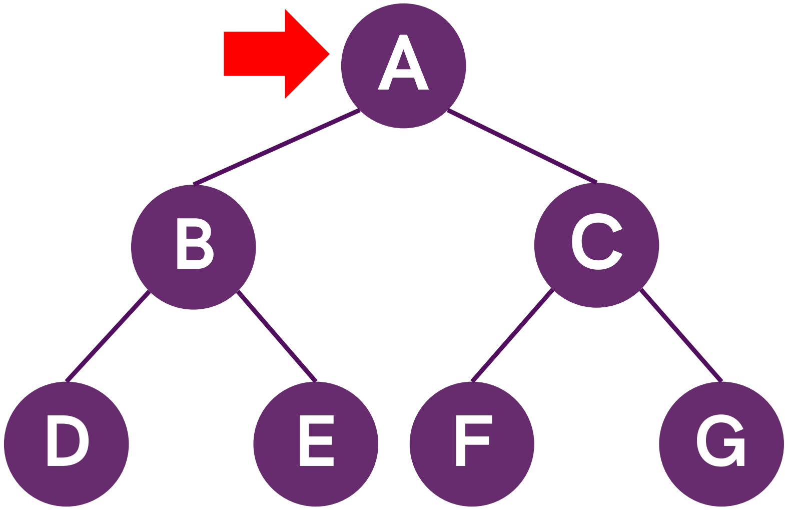


- Visit yourself
- Visit all your left subtree
- Visit all your right subtree

Visited:

A B D E C F G

Pre-Order Traversal

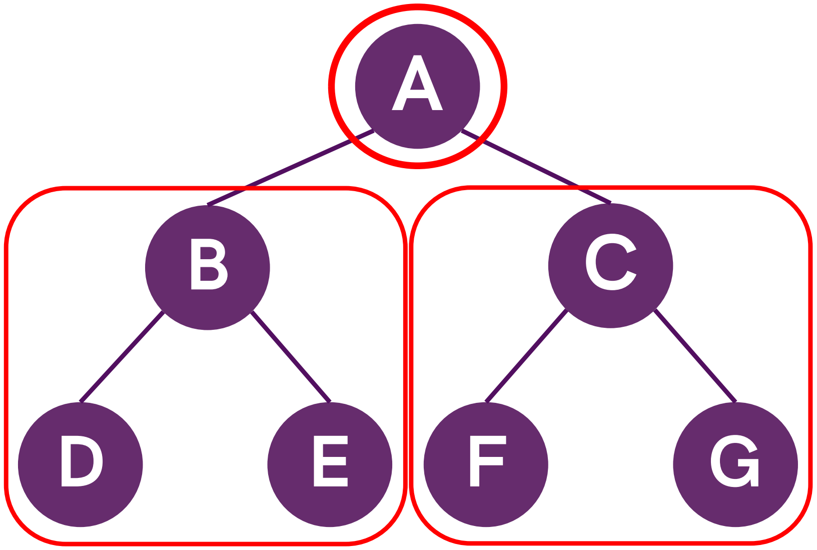


- Visit yourself
- Visit all your left subtree
- Visit all your right subtree

Visited:

A B D E C F G

Pre-Order Traversal

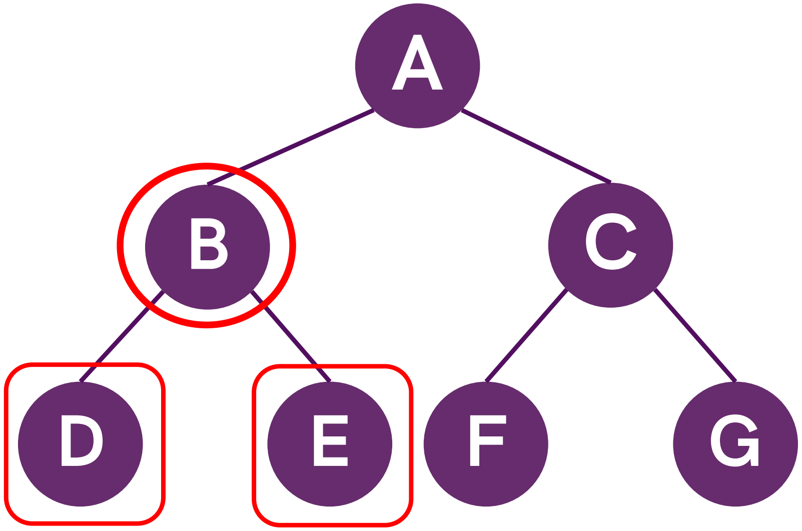


- Visit yourself
- Visit all your left subtree
- Visit all your right subtree

Visited:

A B D E C F G

Pre-Order Traversal

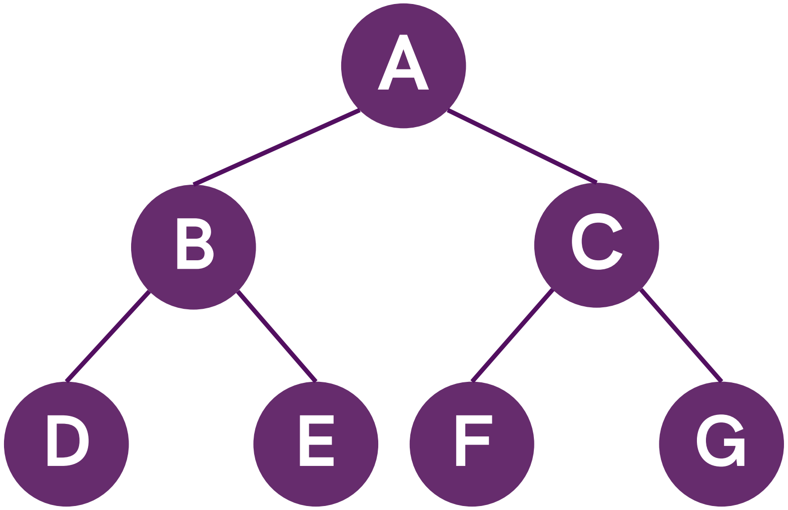


- Visit yourself
- Visit all your left subtree
- Visit all your right subtree

Visited:

A B D E C F G

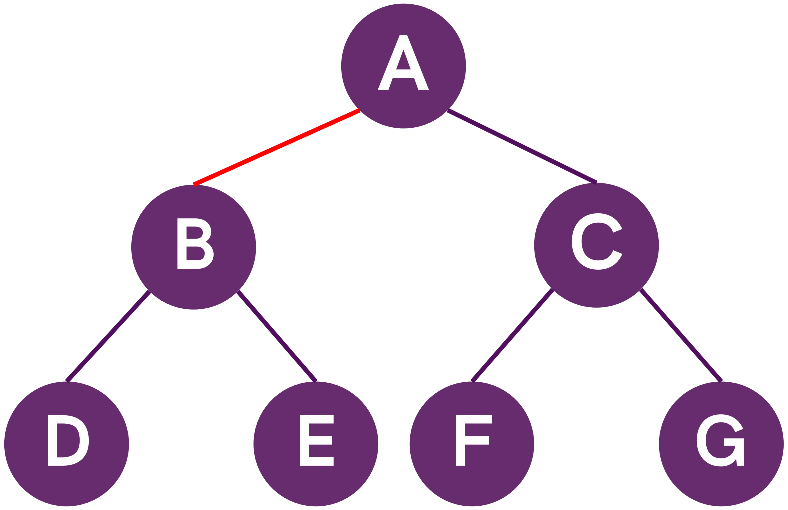
Post-Order Traversal



- Visit all your left subtree
- Visit all your right subtree
- Visit yourself

Visited:

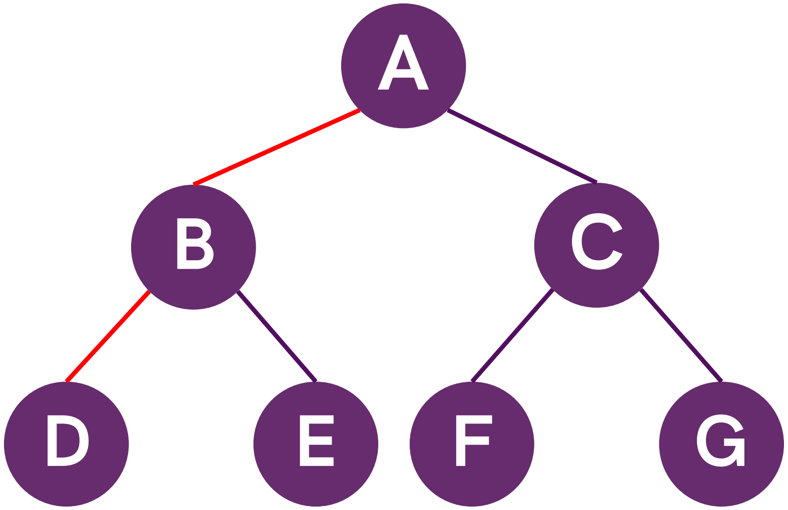
Post-Order Traversal



- Visit all your left subtree
- Visit all your right subtree
- Visit yourself

Visited:

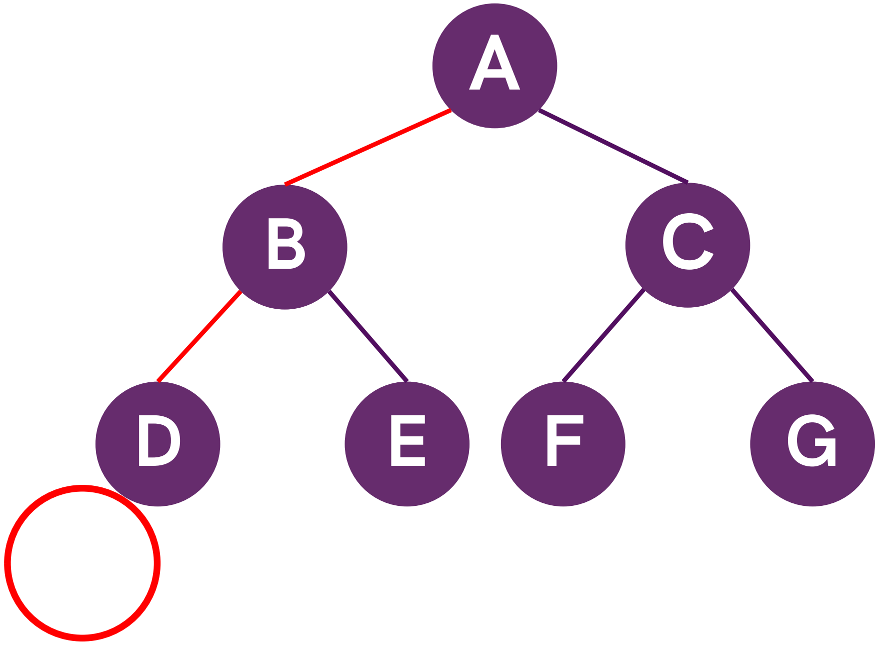
Post-Order Traversal



- Visit all your left subtree
- Visit all your right subtree
- Visit yourself

Visited:

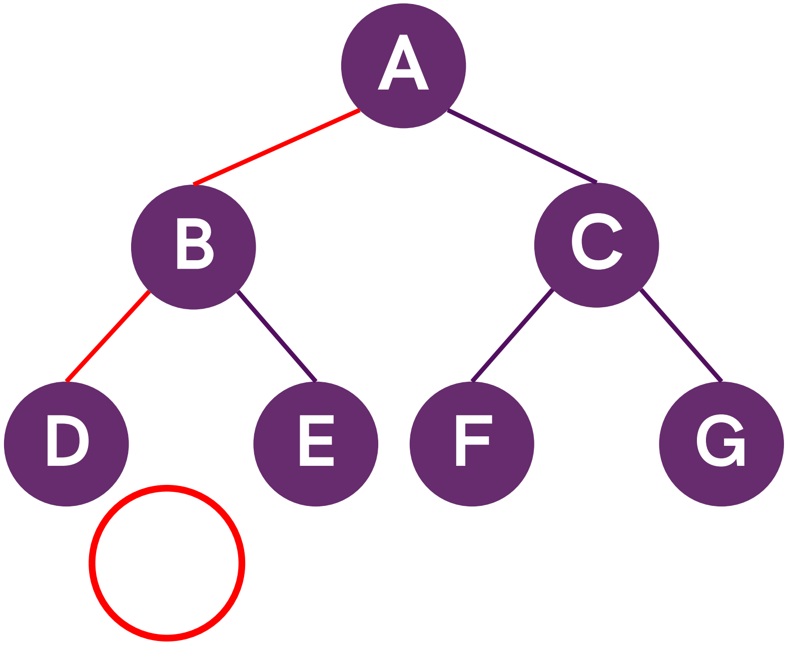
Post-Order Traversal



- Visit all your left subtree
- Visit all your right subtree
- Visit yourself

Visited:

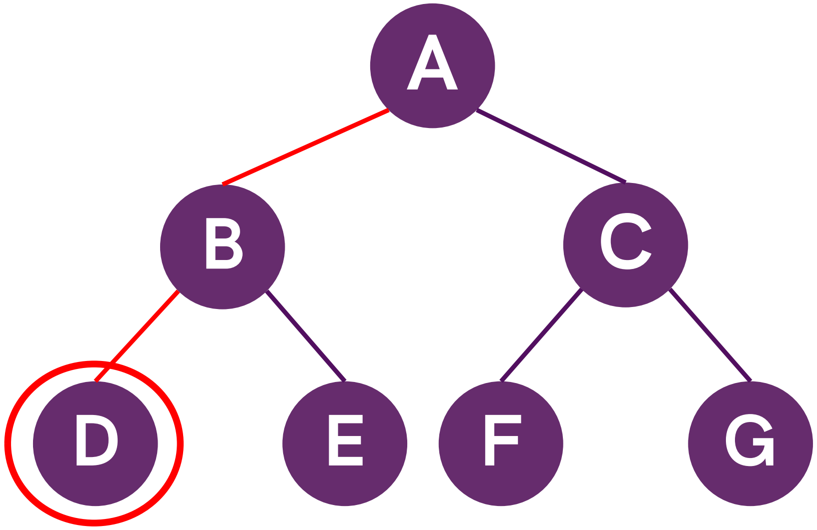
Post-Order Traversal



- Visit all your left subtree
- Visit all your right subtree
- Visit yourself

Visited:

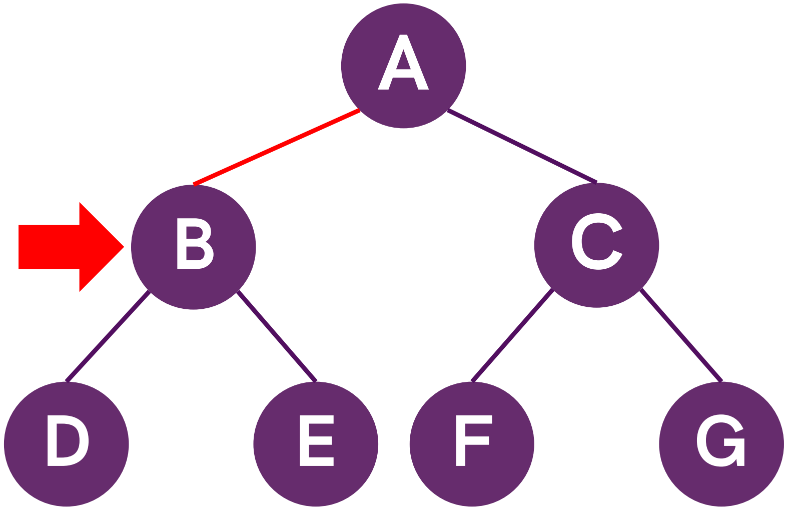
Post-Order Traversal



- Visit all your left subtree
- Visit all your right subtree
- Visit yourself

Visited:
D

Post-Order Traversal

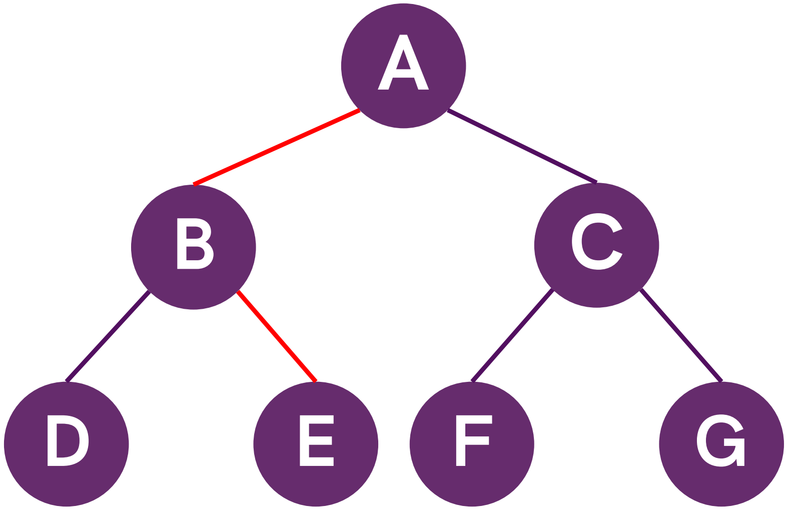


- Visit all your left subtree
- Visit all your right subtree
- Visit yourself

Visited:

D

Post-Order Traversal

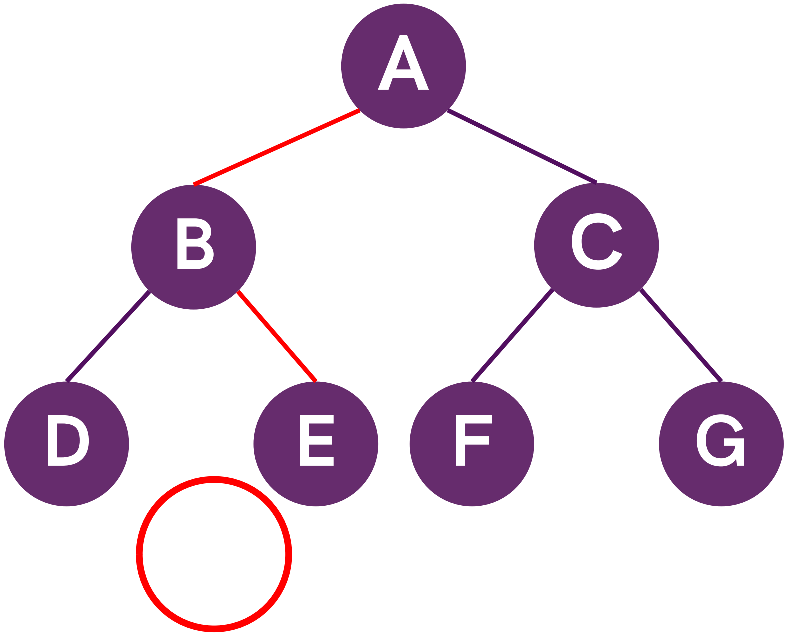


- Visit all your left subtree
- Visit all your right subtree
- Visit yourself

Visited:

D

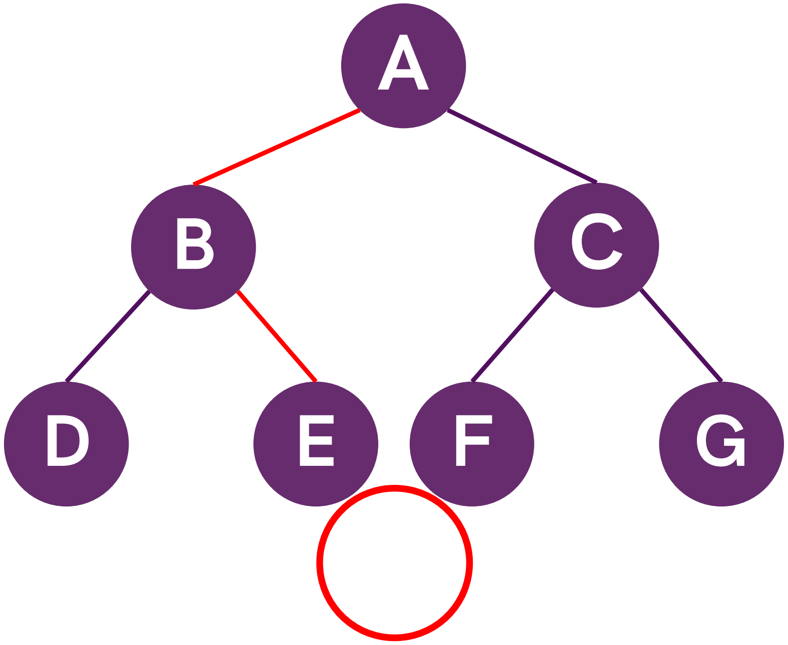
Post-Order Traversal



- Visit all your left subtree
- Visit all your right subtree
- Visit yourself

Visited:
D

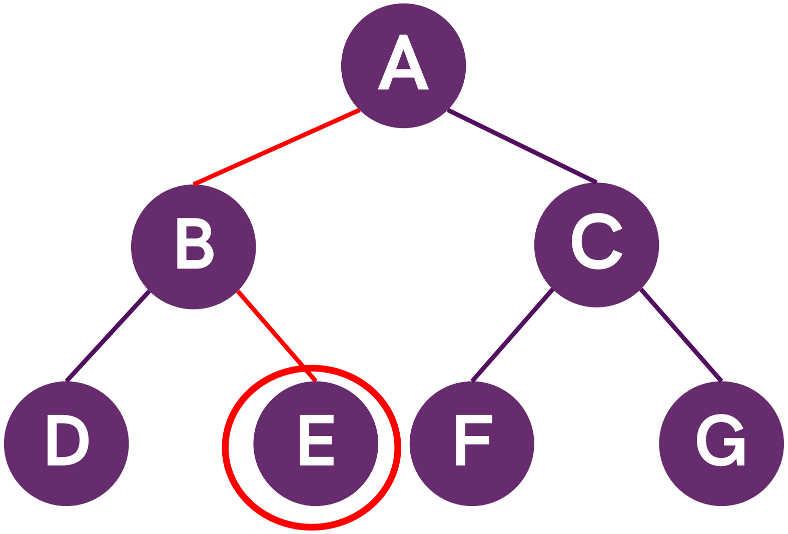
Post-Order Traversal



- Visit all your left subtree
- Visit all your right subtree
- Visit yourself

Visited:
D

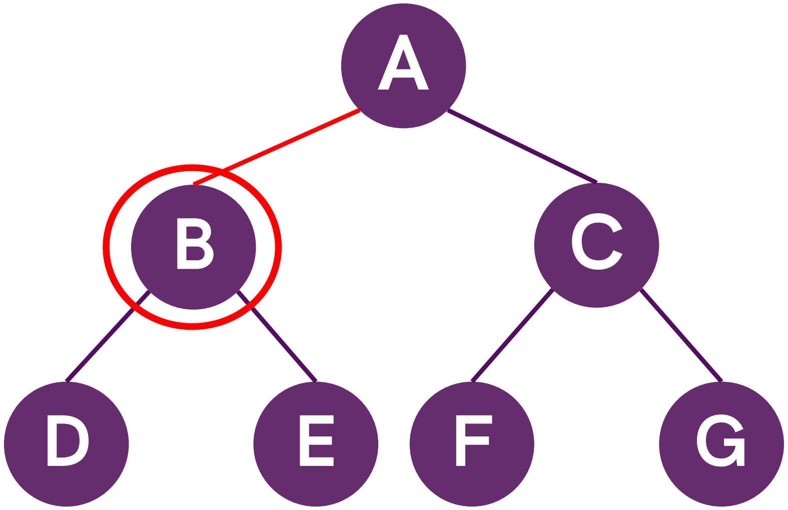
Post-Order Traversal



- Visit all your left subtree
- Visit all your right subtree
- Visit yourself

Visited:
D E

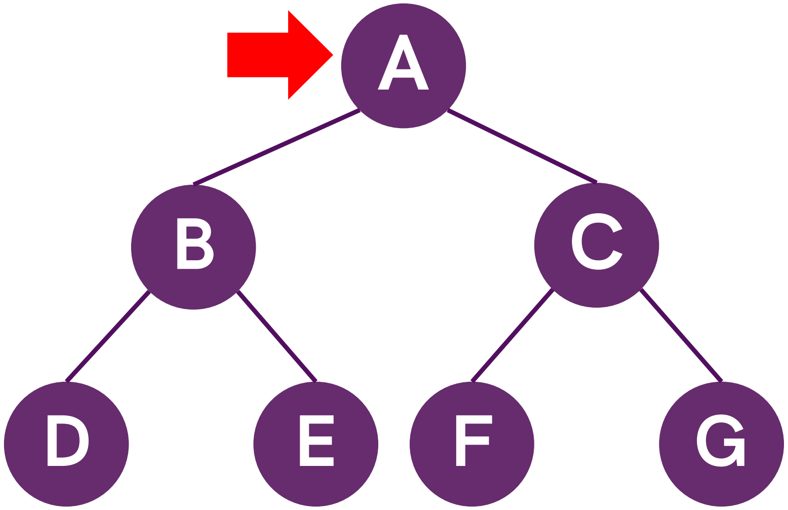
Post-Order Traversal



- Visit all your left subtree
- Visit all your right subtree
- Visit yourself

Visited:
D E B

Post-Order Traversal

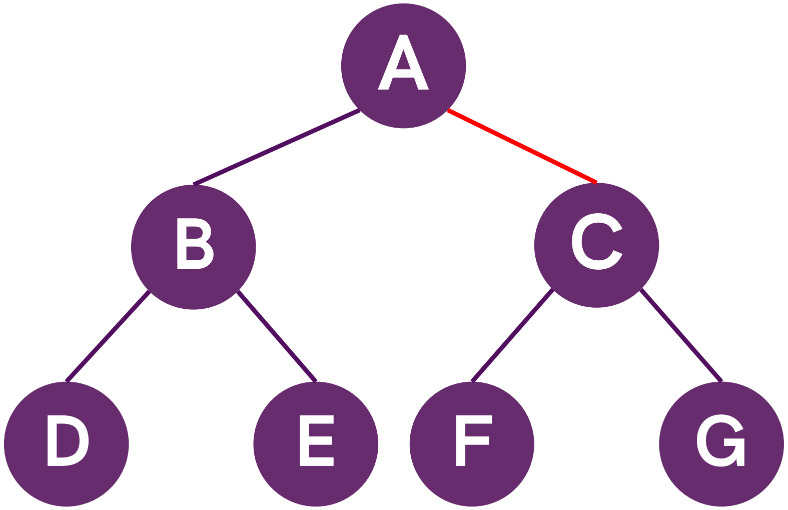


- Visit all your left subtree
- Visit all your right subtree
- Visit yourself

Visited:

D E B

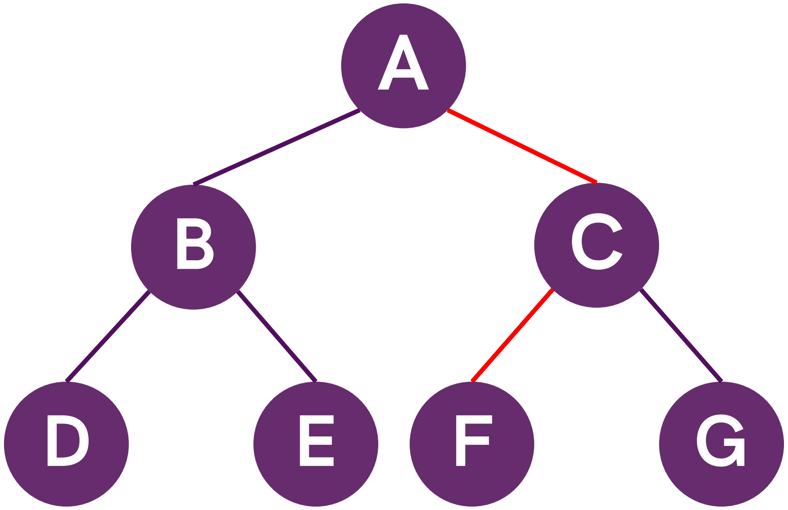
Post-Order Traversal



- Visit all your left subtree
- Visit all your right subtree
- Visit yourself

Visited:
D E B

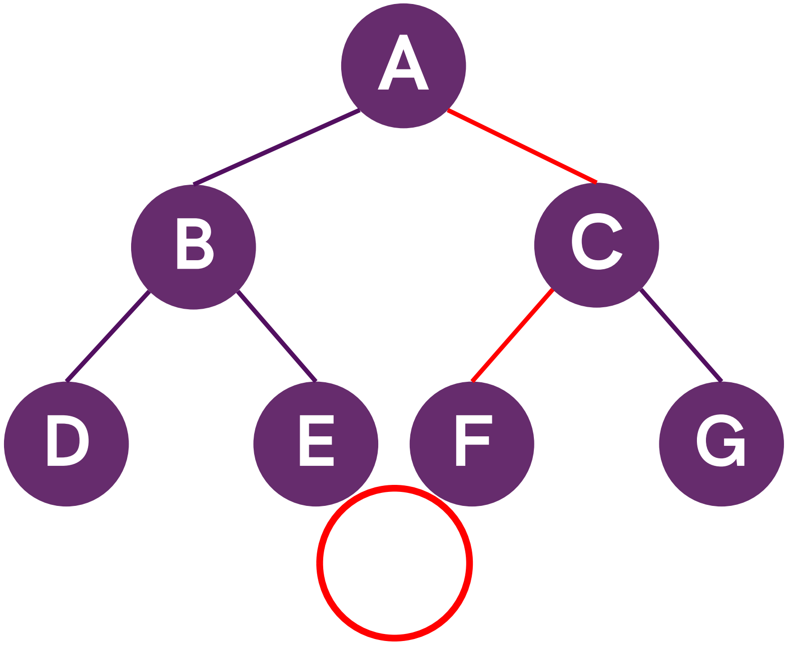
Post-Order Traversal



- Visit all your left subtree
- Visit all your right subtree
- Visit yourself

Visited:
D E B

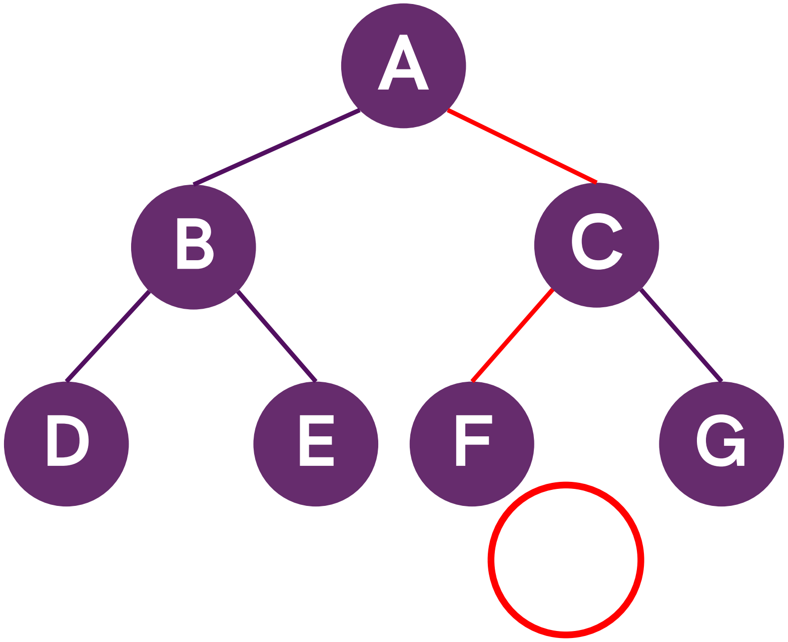
Post-Order Traversal



- Visit all your left subtree
- Visit all your right subtree
- Visit yourself

Visited:
D E B

Post-Order Traversal

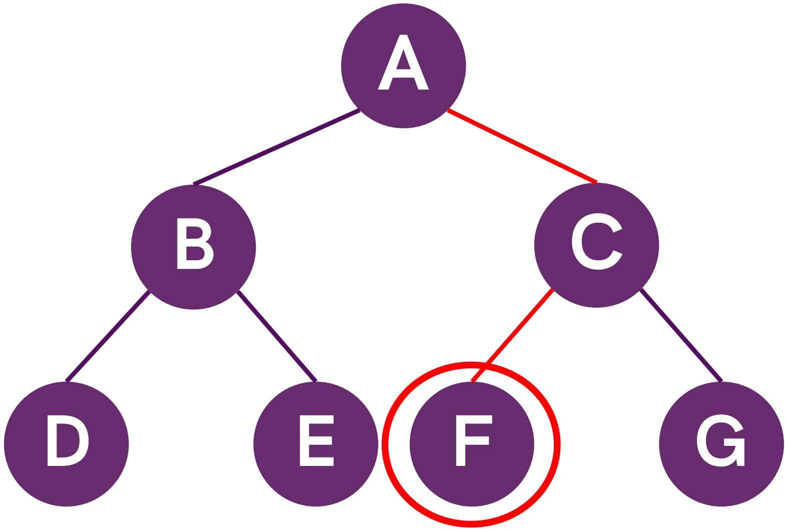


- Visit all your left subtree
- Visit all your right subtree
- Visit yourself

Visited:

D E B

Post-Order Traversal

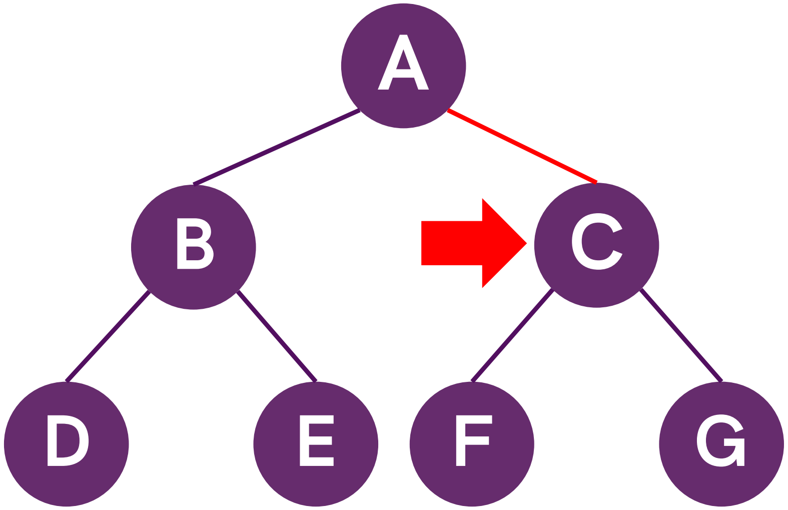


- Visit all your left subtree
- Visit all your right subtree
- Visit yourself

Visited:

D E B F

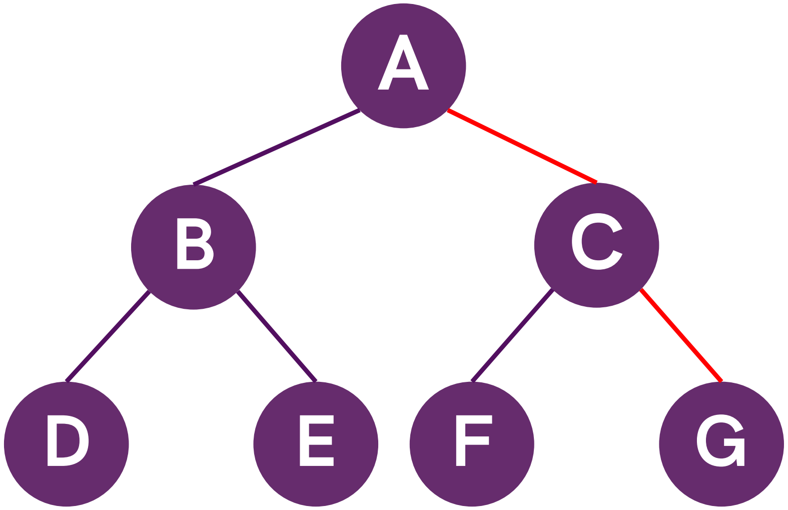
Post-Order Traversal



- Visit all your left subtree
- Visit all your right subtree
- Visit yourself

Visited:
D E B F

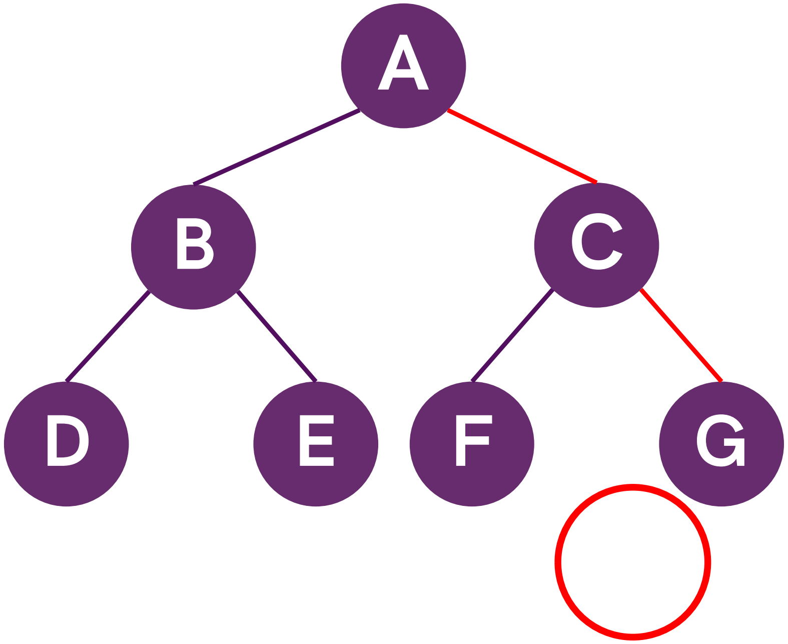
Post-Order Traversal



- Visit all your left subtree
- Visit all your right subtree
- Visit yourself

Visited:
D E B F

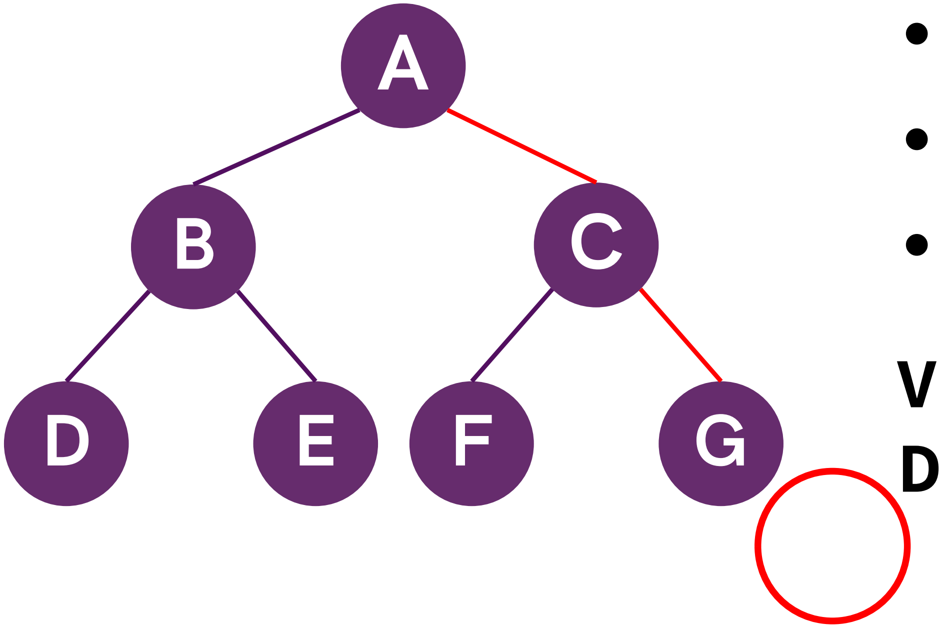
Post-Order Traversal



- Visit all your left subtree
- Visit all your right subtree
- Visit yourself

Visited:
D E B F

Post-Order Traversal

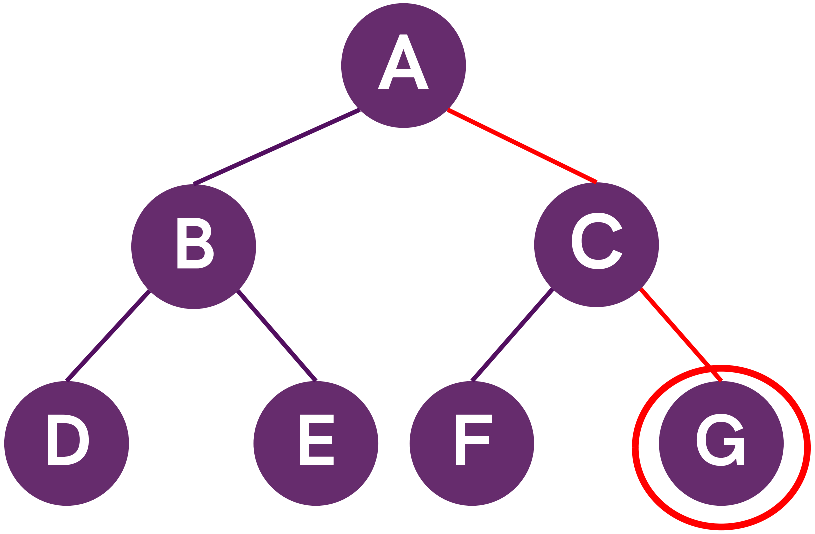


- Visit all your left subtree
- Visit all your right subtree
- Visit yourself

Visited:

D E B F

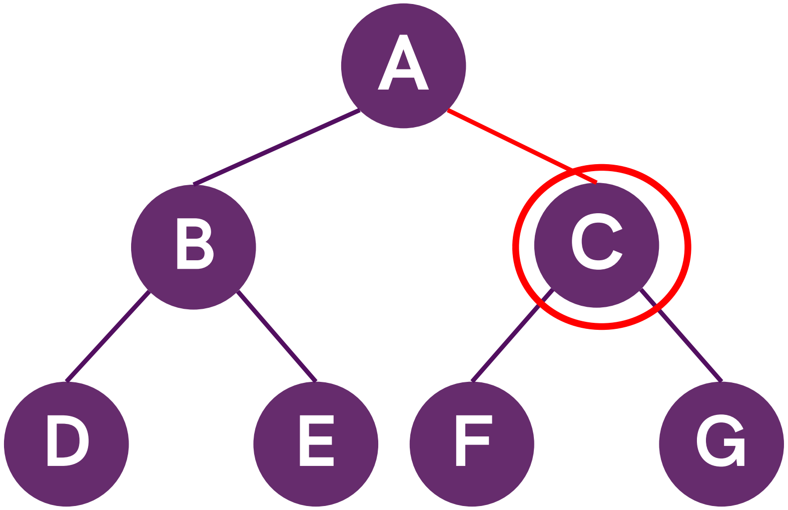
Post-Order Traversal



- Visit all your left subtree
- Visit all your right subtree
- Visit yourself

Visited:
D E B F G

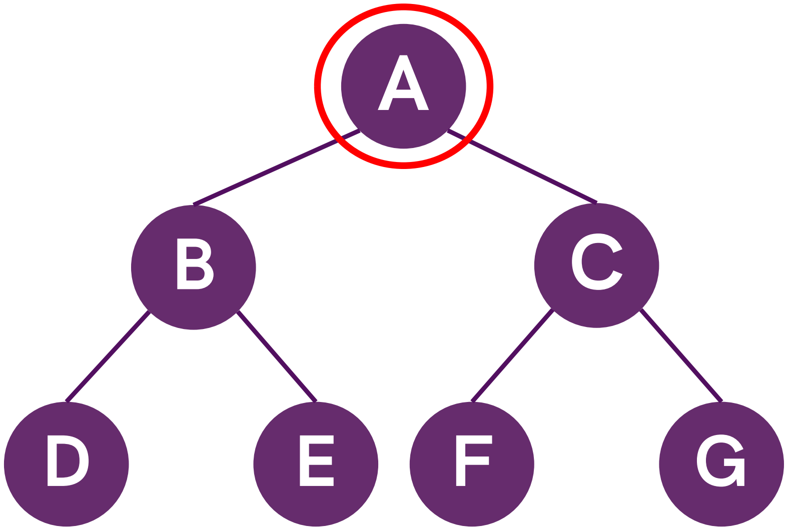
Post-Order Traversal



- Visit all your left subtree
- Visit all your right subtree
- Visit yourself

Visited:
D E B F G C

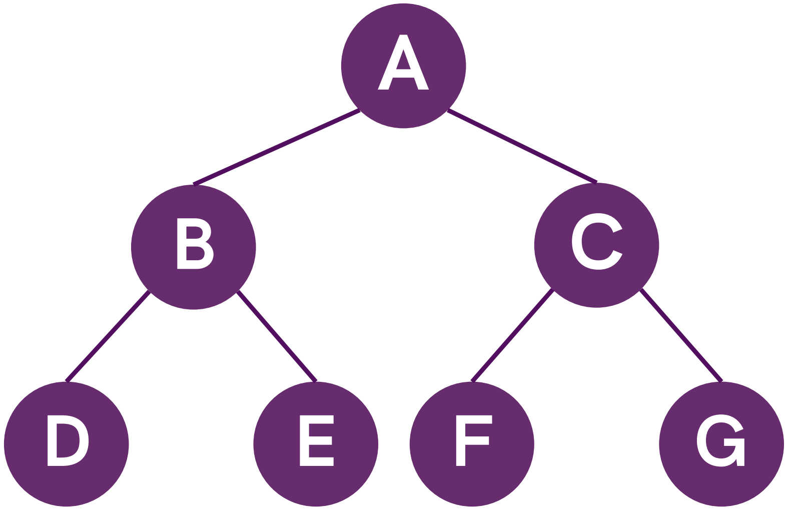
Post-Order Traversal



- Visit all your left subtree
- Visit all your right subtree
- Visit yourself

Visited:
D E B F G C A

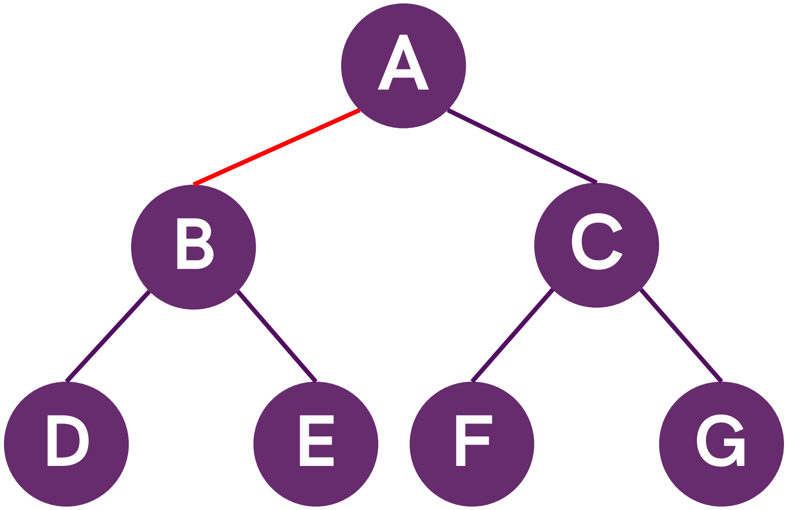
In-Order Traversal



- Visit all your left subtree
- Visit yourself
- Visit all your right subtree

Visited:

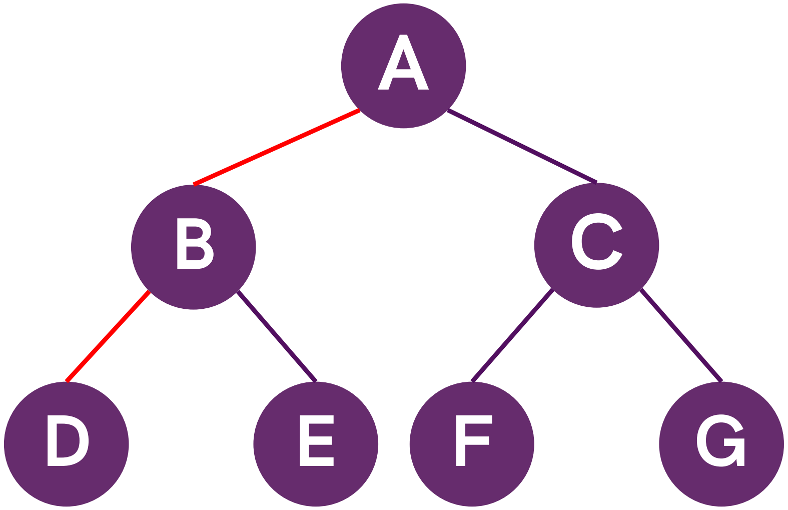
In-Order Traversal



- Visit all your left subtree
- Visit yourself
- Visit all your right subtree

Visited:

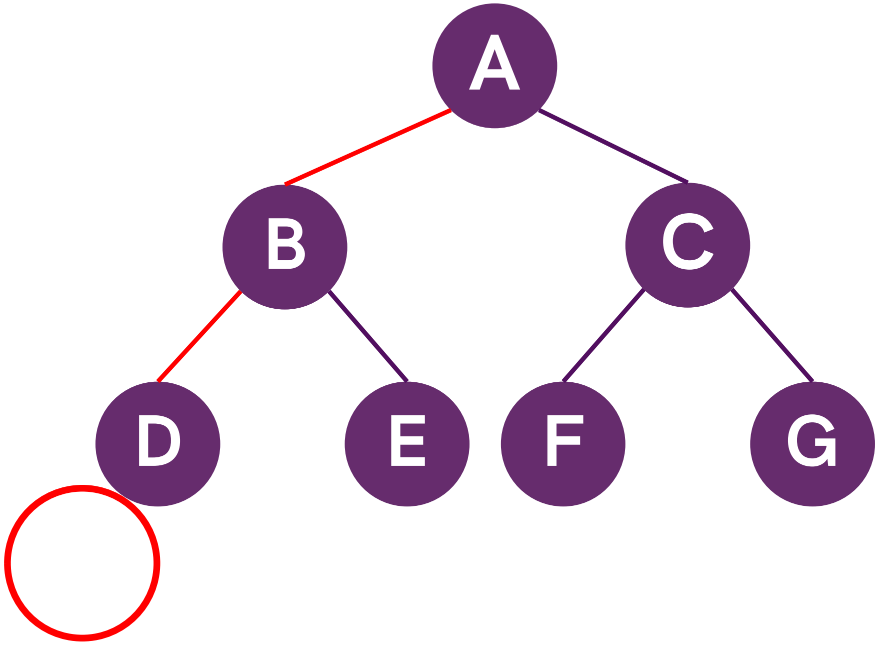
In-Order Traversal



- Visit all your left subtree
- Visit yourself
- Visit all your right subtree

Visited:

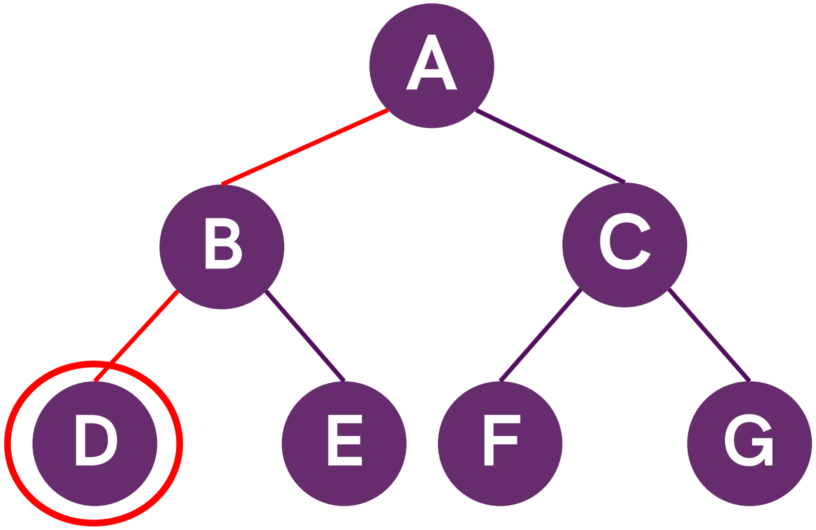
In-Order Traversal



- Visit all your left subtree
- Visit yourself
- Visit all your right subtree

Visited:

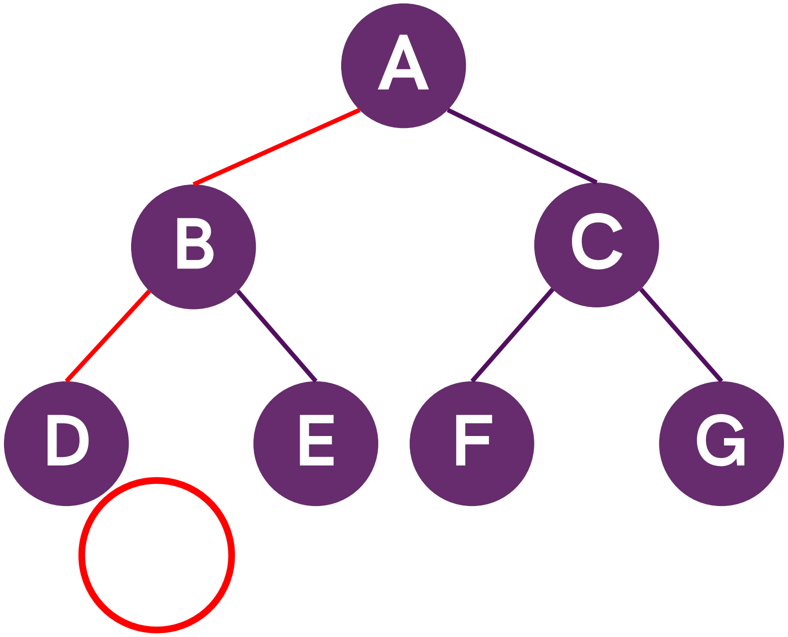
In-Order Traversal



- Visit all your left subtree
- Visit yourself
- Visit all your right subtree

Visited:
D

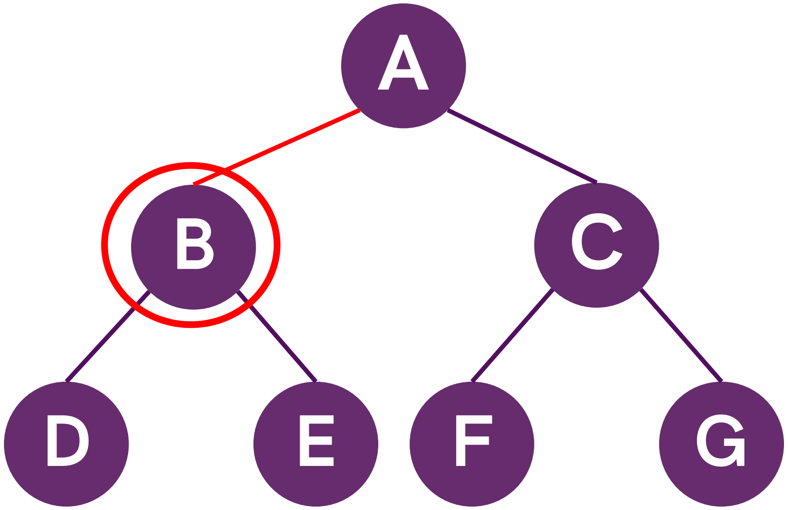
In-Order Traversal



- Visit all your left subtree
- Visit yourself
- Visit all your right subtree

Visited:
D

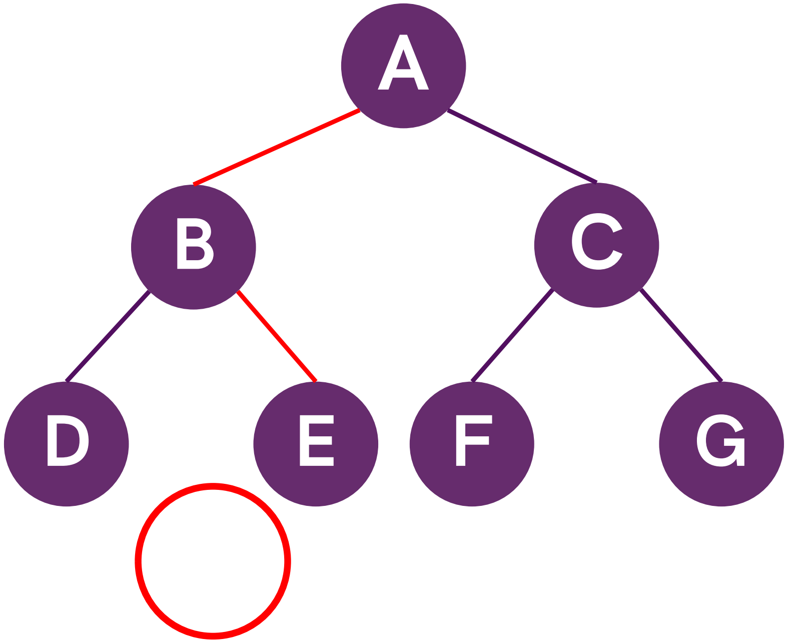
In-Order Traversal



- Visit all your left subtree
- Visit yourself
- Visit all your right subtree

Visited:
D B

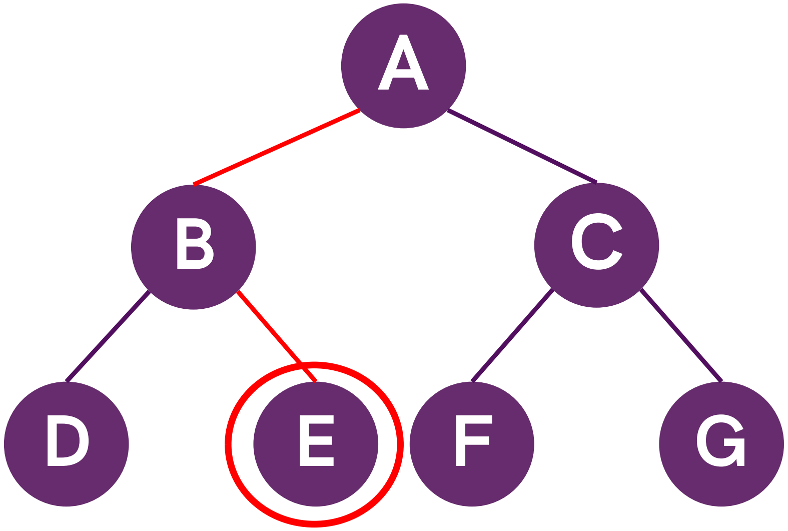
In-Order Traversal



- Visit all your left subtree
- Visit yourself
- Visit all your right subtree

Visited:
D B

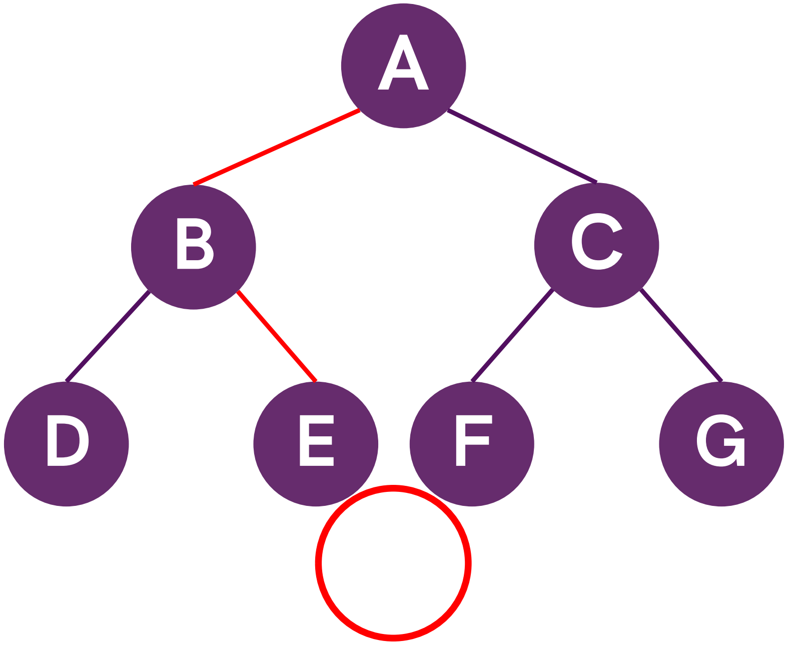
In-Order Traversal



- Visit all your left subtree
- Visit yourself
- Visit all your right subtree

Visited:
D B E

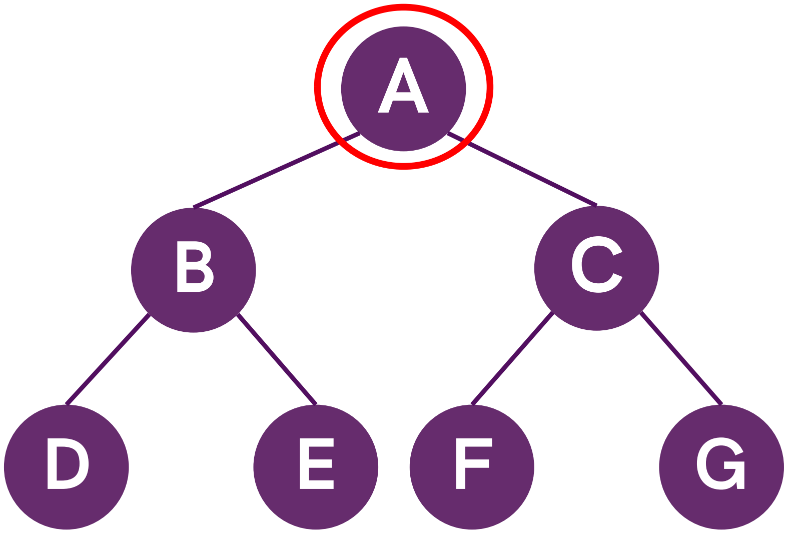
In-Order Traversal



- Visit all your left subtree
- Visit yourself
- Visit all your right subtree

Visited:
D B E

In-Order Traversal

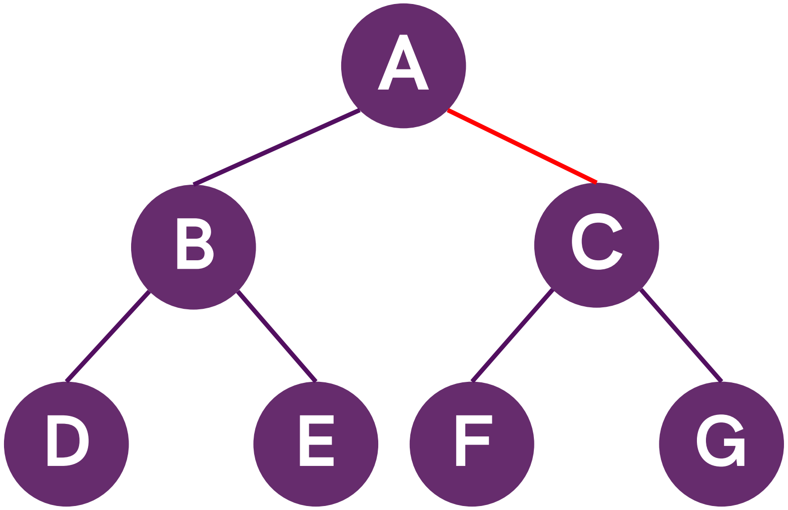


- Visit all your left subtree
- Visit yourself
- Visit all your right subtree

Visited:

D B E A

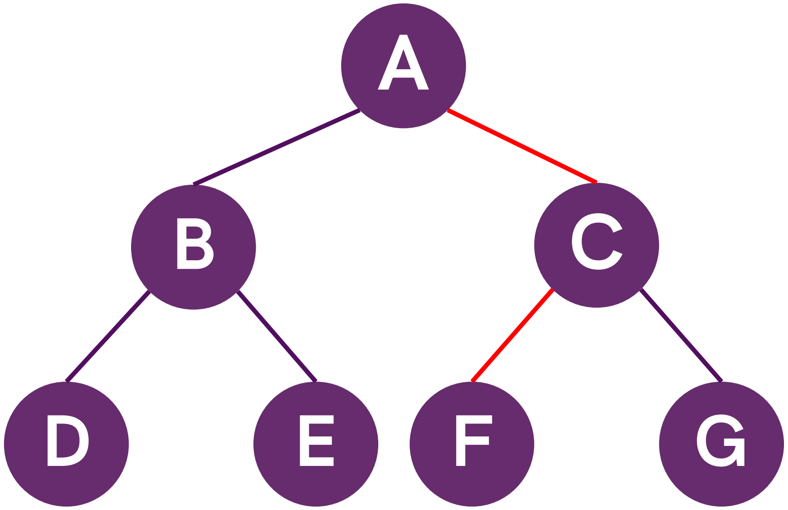
In-Order Traversal



- Visit all your left subtree
- Visit yourself
- Visit all your right subtree

Visited:
D B E A

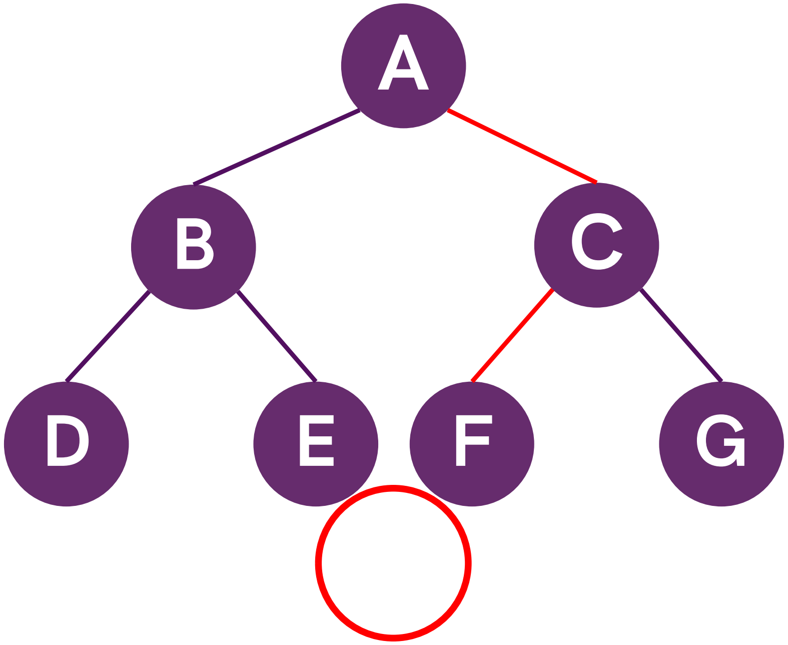
In-Order Traversal



- Visit all your left subtree
- Visit yourself
- Visit all your right subtree

Visited:
D B E A

In-Order Traversal

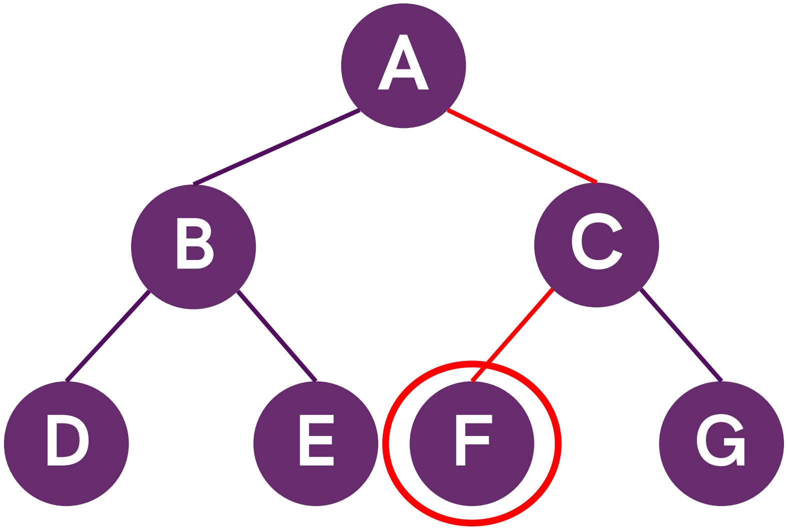


- Visit all your left subtree
- Visit yourself
- Visit all your right subtree

Visited:

D B E A

In-Order Traversal

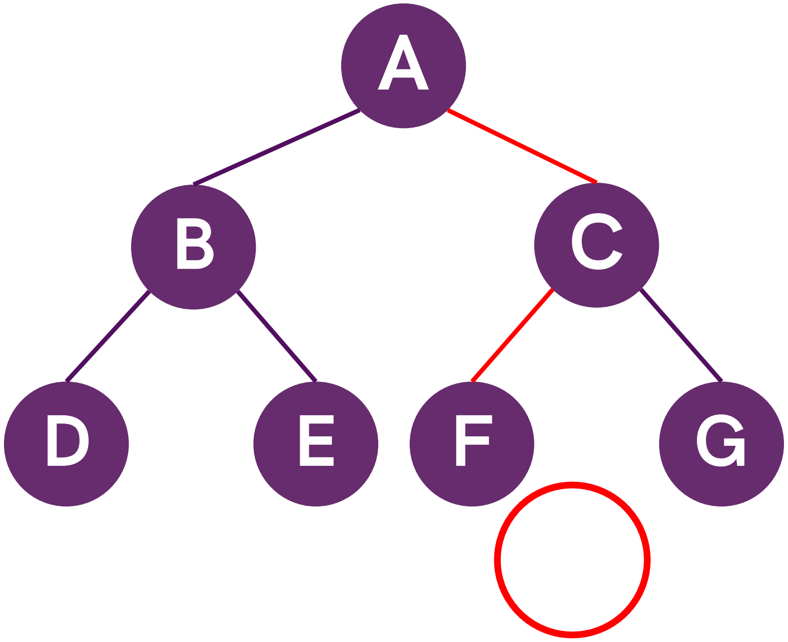


- Visit all your left subtree
- Visit yourself
- Visit all your right subtree

Visited:

D B E A F

In-Order Traversal

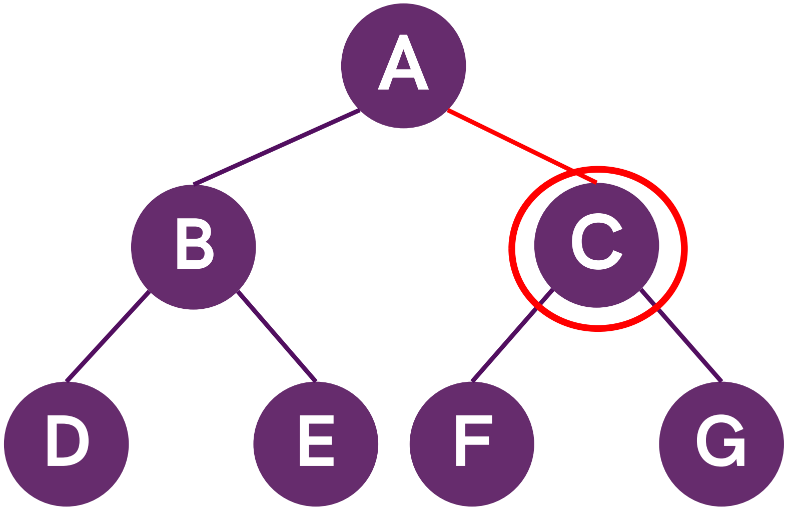


- Visit all your left subtree
- Visit yourself
- Visit all your right subtree

Visited:

D B E A F

In-Order Traversal

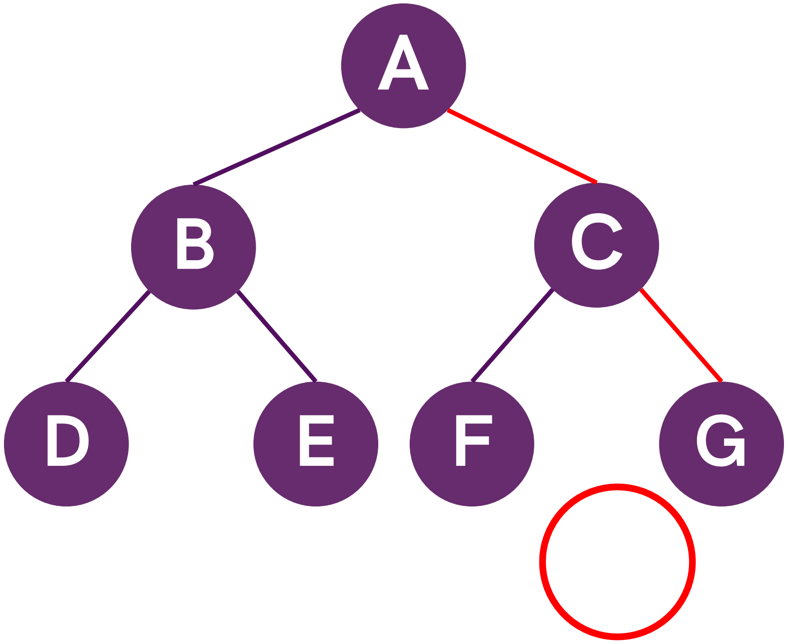


- Visit all your left subtree
- Visit yourself
- Visit all your right subtree

Visited:

D B E A F C

In-Order Traversal

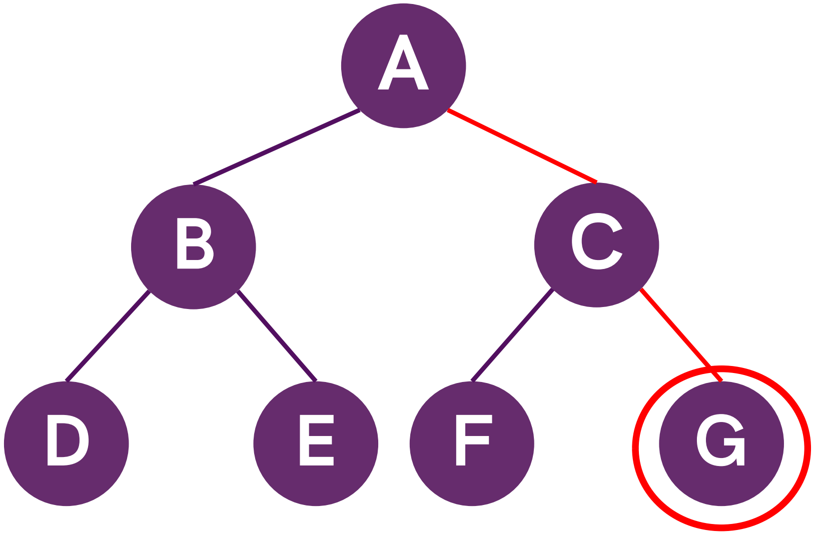


- Visit all your left subtree
- Visit yourself
- Visit all your right subtree

Visited:

D B E A F C

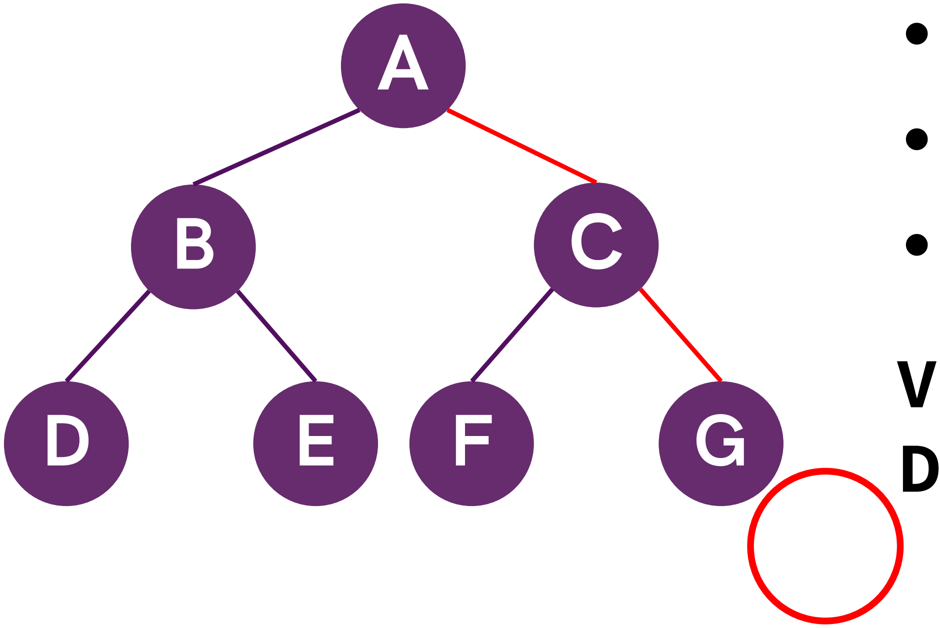
In-Order Traversal



- Visit all your left subtree
- Visit yourself
- Visit all your right subtree

Visited:
D B E A F C G

In-Order Traversal

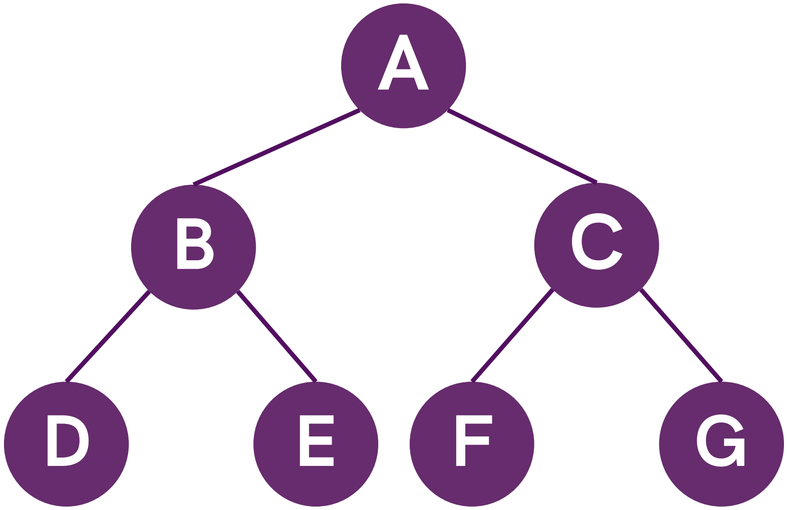


- Visit all your left subtree
- Visit yourself
- Visit all your right subtree

Visited:

D B E A F C G

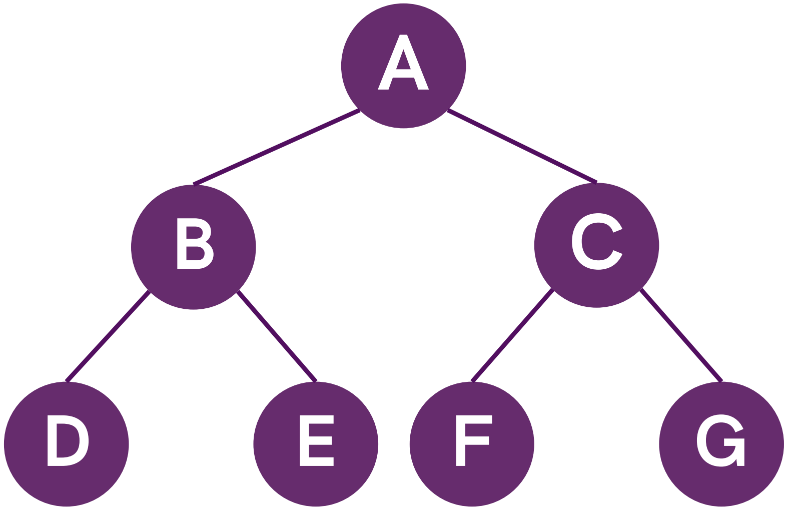
In-Order Traversal



- Visit all your left subtree
- Visit yourself
- Visit all your right subtree

Visited:
D B E A F C G

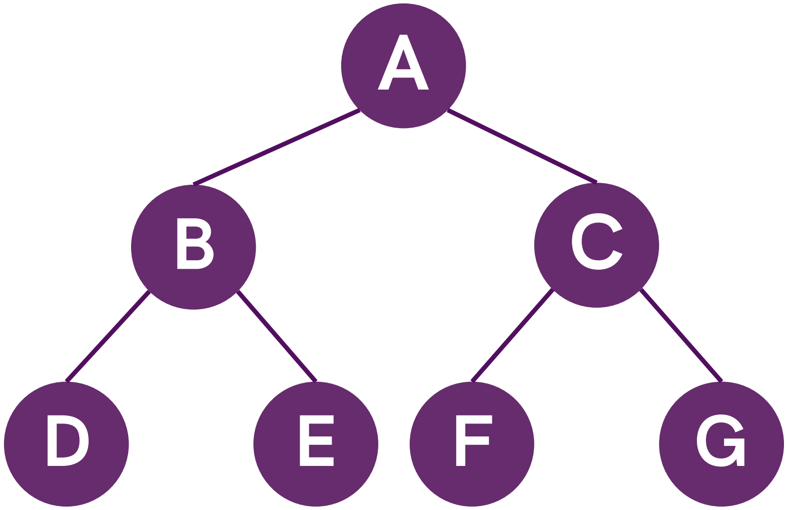
Level-Order Traversal



Visited:
A B C D E F G

Breadth First Traversal

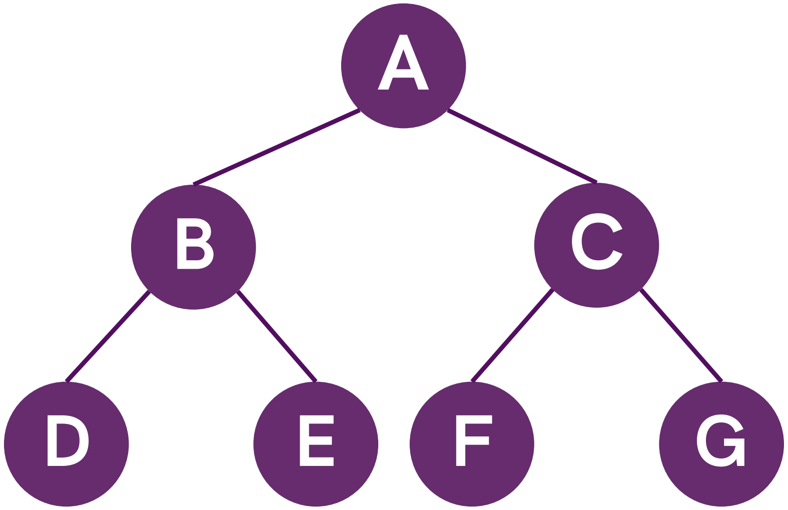
Level-Order Traversal



Visited:
A B C D E F G

Challenging:
When we finish B, how do we know to go to C next?

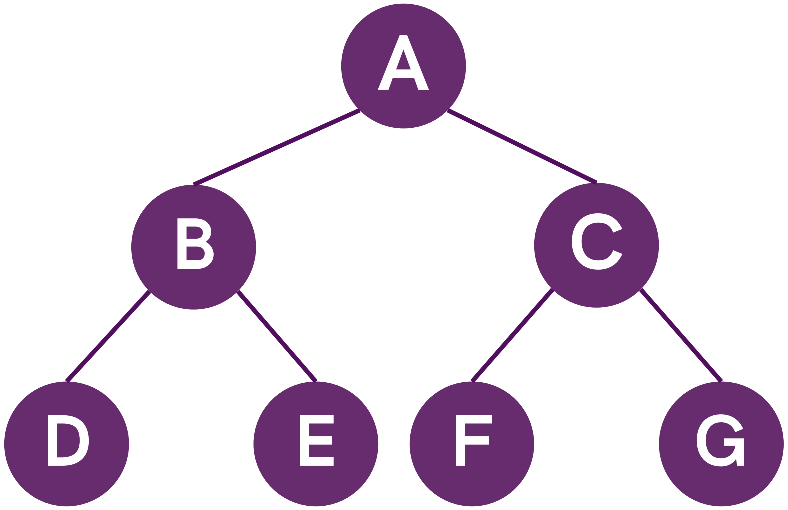
Level-Order Traversal



Visited:
A B C D E F G

Idea:
Keep a list and keep adding to it and removing from start

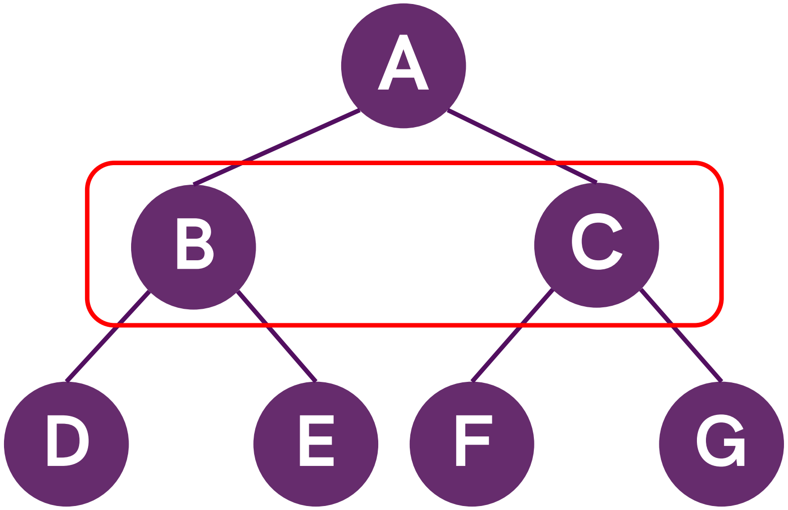
Level-Order Traversal



List:

A

Level-Order Traversal



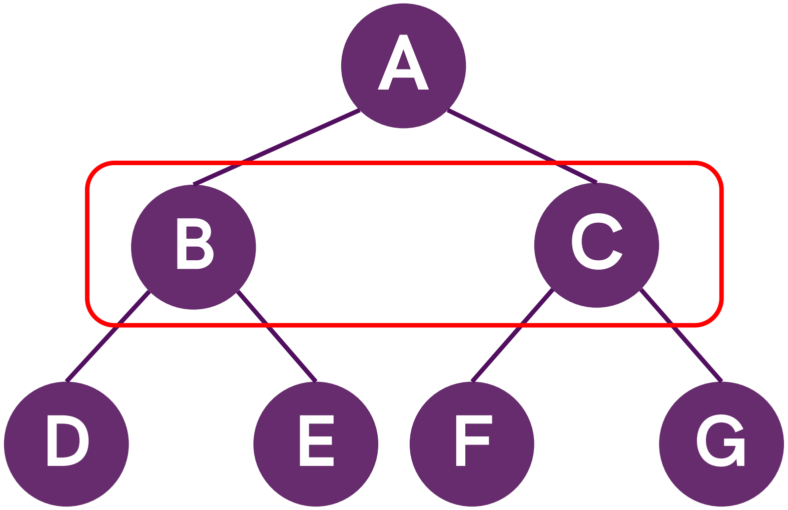
List:

~~A~~

Visited:

A

Level-Order Traversal



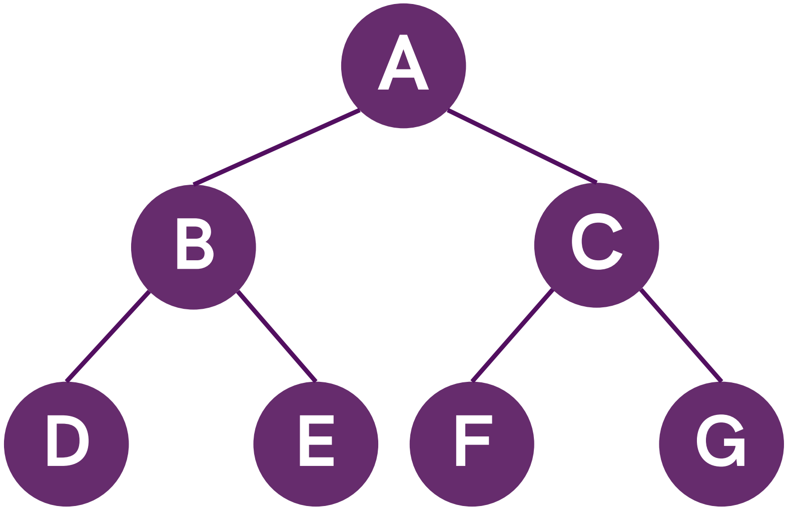
List:

~~A~~ B C

Visited:

A

Level-Order Traversal



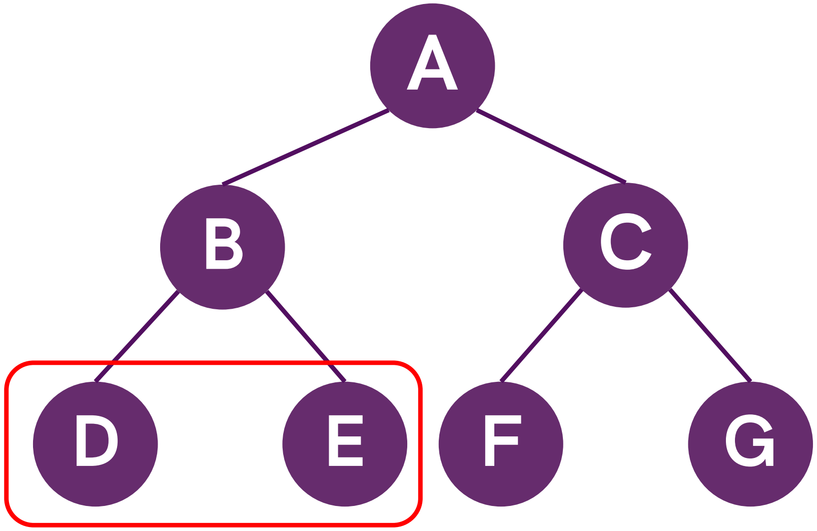
List:

~~A~~ ~~B~~ C

Visited:

A B

Level-Order Traversal



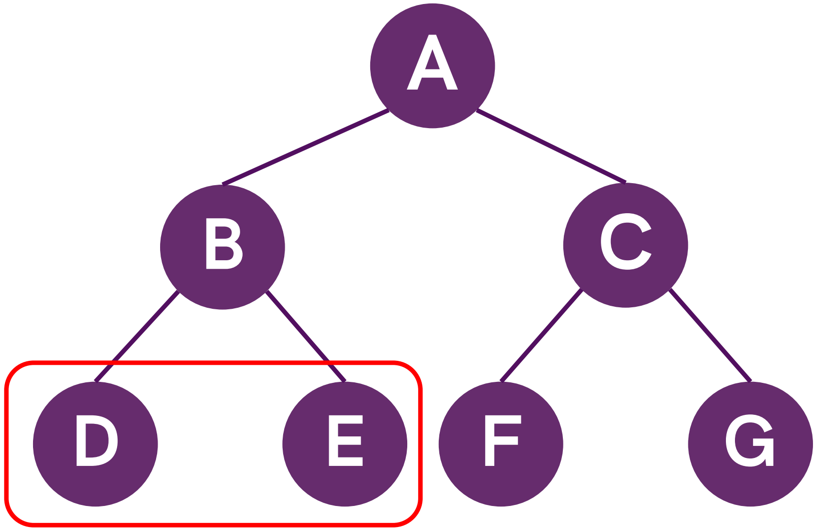
List:

~~A~~ ~~B~~ C

Visited:

A B

Level-Order Traversal



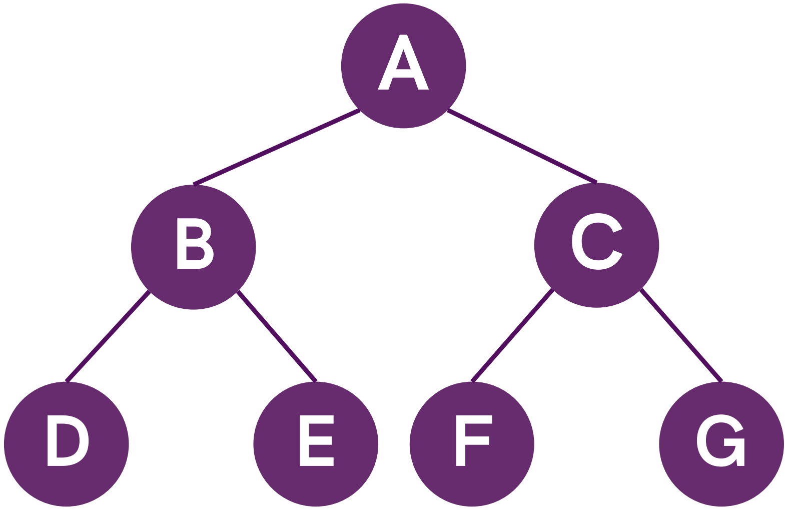
List:

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

Visited:

A B

Level-Order Traversal



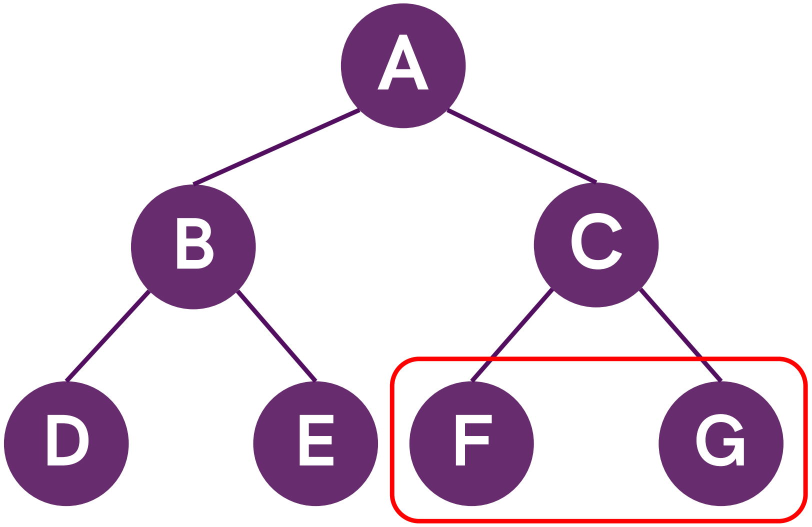
List:

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

Visited:

A B C

Level-Order Traversal



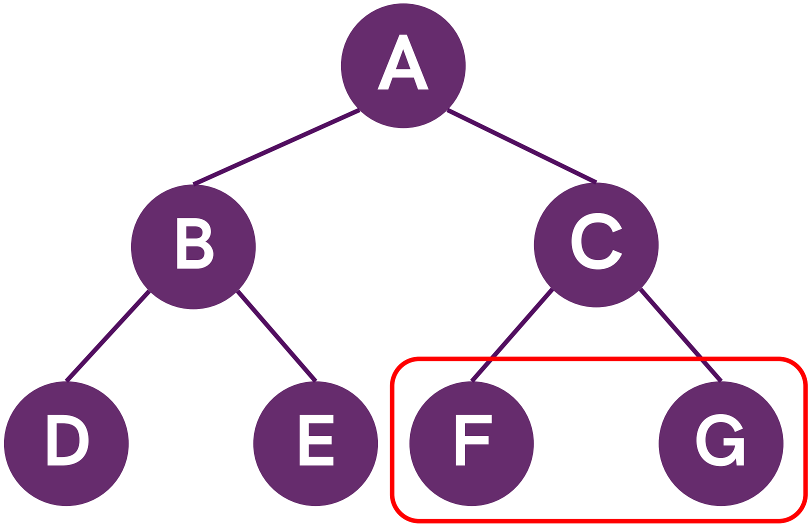
List:

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

Visited:

A B C

Level-Order Traversal



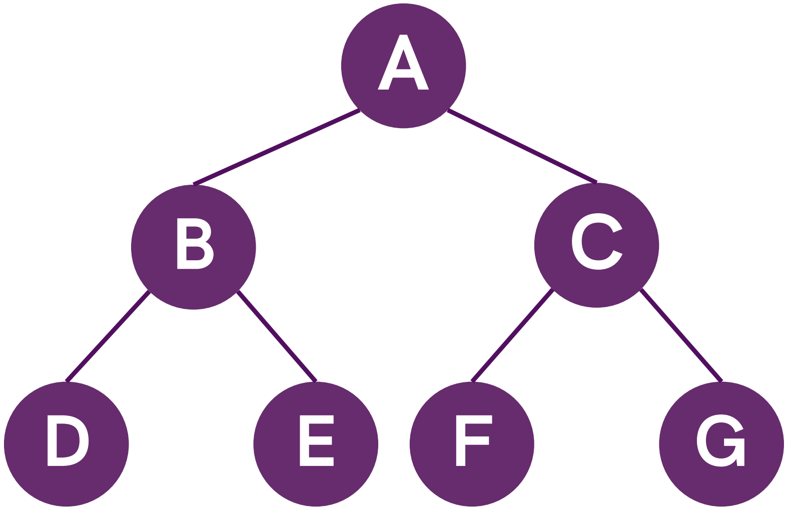
List:

~~A~~ ~~B~~ ~~C~~ D E F G

Visited:

A B C

Level-Order Traversal



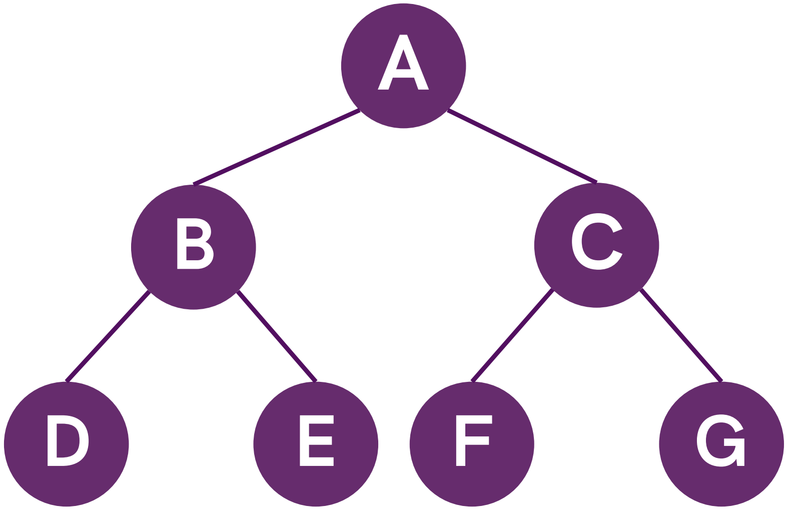
List:

~~A~~ ~~B~~ ~~C~~ ~~D~~ E F G

Visited:

A B C D

Level-Order Traversal



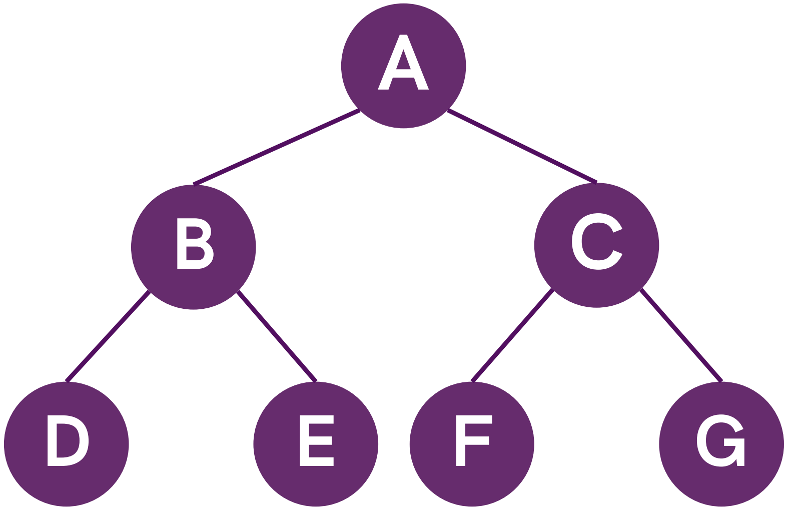
List:

~~A~~ ~~B~~ ~~C~~ ~~D~~ E F G

Visited:

A B C D

Level-Order Traversal



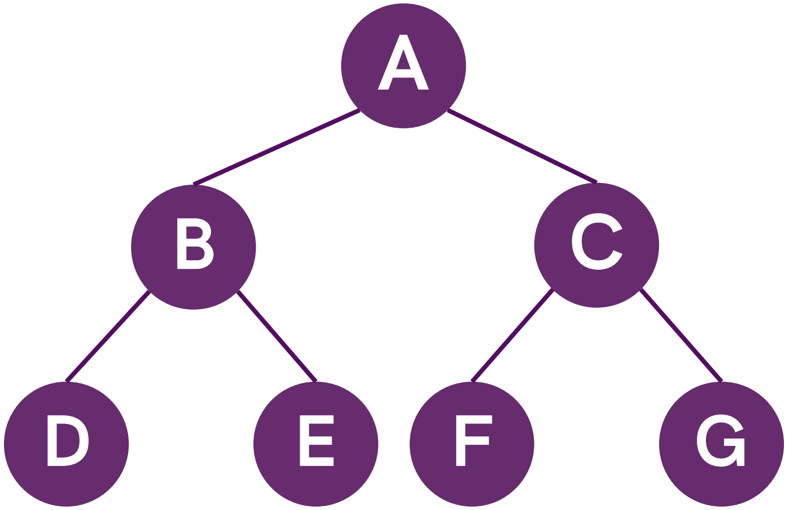
List:

~~A~~ ~~B~~ ~~C~~ ~~D~~ ~~E~~ F G

Visited:

A B C D E

Level-Order Traversal



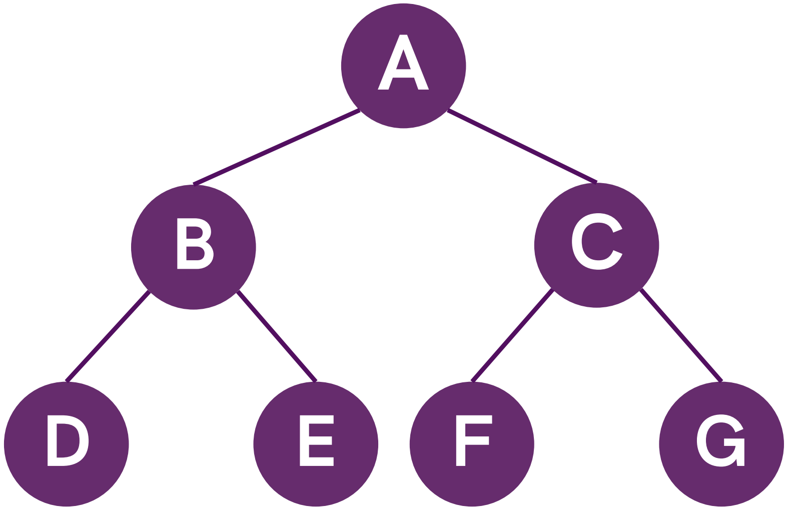
List:

~~A~~ ~~B~~ ~~C~~ ~~D~~ ~~E~~ ~~F~~ G

Visited:

A B C D E F

Level-Order Traversal



List:

~~A B C D E F G~~

Visited:

A B C D E F G

Queue