

Tình huống PHP

(giúp học viên ôn tập lại các kiến thức về Lập trình web thông qua một số ứng dụng thường gặp trong thực tế)

Kết quả Trắc nghiệm trực tuyến

1. Cho biết dung lượng bộ nhớ trong tối đa của một CPU tổ chức bộ nhớ theo BYTE và có 16 bit địa chỉ ?

- 16KB
- 16MB ✗
- 64MB
- 64KB

2. Trong soạn thảo văn bản Word, công dụng của tổ hợp phím Ctrl - H là:

- Tạo tệp văn bản mới
- Chức năng thay thế trong soạn thảo ✓
- Định dạng chữ hoa
- Lưu tệp văn bản vào đĩa

3. Chức năng chính của tập các thanh ghi (Registers) là?

- Điều khiển nhận lệnh
- Giải mã lệnh và thực thi lệnh ✗
- Vận chuyển thông tin giữa các thành phần bên trong máy tính
- Chứa các thông tin phục vụ cho hoạt động của CPU

4. Trong khi làm việc với Word, tổ hợp phím tắt nào cho phép chọn tất cả văn bản đang soạn thảo:

- Ctrl + A
- Alt + A ✗
- Alt + F
- Ctrl + F

5. Khi đang soạn thảo văn bản Word, muốn phục hồi thao tác vừa thực hiện thì bấm tổ hợp phím:

- Ctrl - Z ✓
- Ctrl - X
- Ctrl - V
- Ctrl - Y

Làm lại

Kỹ thuật hiển thị dữ liệu bằng ajax



The Face Reader

2013



The Pirate Fairy

2014



Sadako

2013



Resident Evil 3

2007

Xem thêm

Kỹ thuật phân trang bằng ajax

Khoa học trực tuyến



Khóa học lập trình jQuery (jQuery Master)

Nếu nói đến lập trình web, chúng ta nghĩ ngay đến một ngôn ngữ lập trình không thể thiếu trong một website đó là Javascript. Javascript giúp chúng ta thực hiện những thao tác, hiệu ứng mà các ngôn ngữ lập web như PHP, ASP, JSP... khó mà thực hiện được.



Xây dựng web chat với NodeJS

Khóa học jQuery này cũng bổ sung thêm cho các bạn một Javascript framework khác đó là NodeJS qua một ví dụ thực tế là ứng dụng Web chat. Với kiến thức và ví dụ của NodeJS các bạn có thể dễ dàng làm phần Chat support trên website của chính mình



Khóa học lập trình PHP

PHP đã và đang trở thành một ngôn ngữ lập trình phổ biến trong các ứng dụng web ngày nay. Với các ưu điểm như miễn phí, dễ đọc, thao tác tốt với các hệ quản trị cơ sở dữ liệu, cộng đồng sử dụng rộng rãi, thư viện phong phú ... PHP ngày càng phát triển và việc học PHP đã trở nên đặc biệt cần thiết đối với bất kỳ lập trình

Page 2 of 3

« Previous

Page 2

▼

Next »

Tổng hợp tin tức thể thao - RSS

VnExpress

DanTri



Ông Kiếm 'Hâm' và hàng chục học trò là nữ tuyển thủ quốc gia

Sống ở miền quê thuần nông nhưng hơn 20 năm nay cựu binh Dương Khắc Kiêm gắn bó với sân bóng nhiều hơn những mảnh ruộng. Trong hàng trăm học trò qua tay ông, rất nhiều người là tuyển thủ quốc gia các cấp độ.



Việt Nam - Thái Lan: Đòi nợ để đi World Cup

Tuyển bóng đá nữ Việt Nam sẽ giành vé dự ngày hội lớn của bóng đá thế giới năm 2015 nếu vượt qua kình địch Thái Lan trong trận play-off diễn ra lúc 17h15 chiều nay 21/5.



Tuyển nữ Việt Nam vững tin trước trận tranh vé dự World Cup

Những cô gái vàng của bóng đá Việt Nam đã có buổi tập cuối cùng chuẩn bị cho trận play-off với Thái Lan ngày 21/5.



Tuyển nữ Việt Nam và những cuộc đụng độ Thái Lan

Tính đến trước trận tranh vé vớt dự World Cup 2015 hôm nay 21/5, tuyển nữ Việt Nam đang nhỉnh hơn đôi thủ về số trận thắng trong tổng cộng 15 cuộc đối đầu.

Tổng hợp tin tức thể thao - RE

VnExpress

DanTri



Trọng tài lo tuyển nữ bị ép trong trận gặp Thái Lan

Trước trận play-off tranh vé dự World Cup 2015, trọng tài từng ba lần giành danh hiệu "Còi vàng" Dương Văn Hiền có lý do để băn khoăn về chuyện người cầm cân nảy mực.



Costa cầu cứu 'thần y' Marijuana

Tiền đạo của Atletico Madrid đã đến Serbia hôm qua để gặp nữ bác sĩ Marijana Kovacevic - người sở hữu phương thuốc đặc biệt chiết xuất từ dạ con phụ nữ sau sinh để điều trị chấn thương.



Những màn nội chiến ở chung kết Cup châu Âu

Trận derby Atletico - Real Madrid thứ bảy này là lần thứ năm Champions League chứng kiến hai đội cùng quốc gia tranh ngôi vô địch.



Pirlo say khướt trong lễ rước Cup của Juventus

CLB thành Torino vừa thuân túm chức vô địch Serie A thứ ba liên tiếp và thứ 30 trong lịch sử vẻ vang.

Ứng dụng của tập tin .htaccess

Kiến thức tổng quan

- .htaccess là một file cấu hình có phần mở rộng là htaccess, nó có thể thay đổi các cấu hình do Apache quản lý (httpd.conf) và có tác dụng trong thư mục chứa nó và các thư mục con của thư mục chứa nó.
- .htaccess có một sự ảnh hưởng rất lớn đến sự hoạt động của website do đó cần đặc biệt quan tâm đến file cấu hình này về vấn đề bảo mật, lỗi cú pháp, sự ảnh hưởng, ...
- Apache sẽ tìm và thực thi tất cả những file .htaccess , do vậy sẽ ảnh hưởng đến tốc độ của website
→ chỉ nên dùng .htaccess trong trường hợp bạn không có quyền cấu hình trên httpd.conf của Apache

Ứng dụng của tập tin .htaccess

1. Hạn chế IP truy cập vào website
 2. Điều hướng thông báo lỗi
 3. Thiết lập tập tin thực thi mặc định
 4. Hạn chế view thư mục
 5. Hạn chế truy cập vào tập tin
 6. Tạo mật khẩu cho thư mục tập tin
 7. Chống chia sẻ tài nguyên
 8. Thiết lập kích thước tập tin upload
 9. Rewrite URL
- ...

Lập trình hướng đối tượng chuyên sâu

(cung cấp kiến thức lập trình hướng đối tượng và tiếp cận với một số khái niệm mới trong lập trình hướng đối tượng)

Ôn tập Lập trình hướng đối tượng

- | | |
|---------------------------------|-----------------------------|
| 1. Class & Object | 9. constant |
| 2. Property & Method | 10. clone() |
| 3. Constructor & Destructor | 11. __autoload() |
| 4. Extends & Override | 12. spl_autoload_register() |
| 5. Final | 13. __sleep() & __wakeup() |
| 6. Public – Private – Protected | 14. toString() |
| 7. Static | 15. __set() & __get() |
| 8. Parent & Self | |

Abstract Class and Interface in PHP

Abstract Class

- Abstract Class (Lớp trừu tượng) là một Class dùng để định nghĩa các thuộc tính và phương thức chung cho các class khác
- Abstract Class không cho phép khởi tạo đối tượng từ class đó.
- Abstract method không cho phép định nghĩa rõ cách hoạt động của method, chỉ dùng lại ở khai báo method
- Từ khóa abstract được dùng để định nghĩa một Lớp trừu tượng.

Abstract Class and Interface in PHP

Interface

- Interface không phải là class, đây là một mẫu giao diện quy định thêm một số thuộc tính và phương thức cho một class nào đó
- Interface không cho phép định nghĩa rõ cách hoạt động của method, chỉ dùng lại ở khai báo method
- Việc định nghĩa các method này sẽ được thực hiện ở class con

Abstract Class and Interface in PHP

- *Abstract class là một class*  *Interface không là một class*
- Abstract class mỗi method không bắt buộc phải được định nghĩa ở lớp con 
Interface mỗi method bắt buộc phải được định nghĩa ở lớp con
- Abstract class mỗi lớp con chỉ kế thừa từ một class abstract, một class abstract có nhiều class con  Interface mỗi lớp con kế thừa từ một hoặc nhiều Interface, một Interface có nhiều class con
- Abstract class mỗi phương thức abstract có thể ở trạng thái protected hoặc public  Interface mỗi phương thức chỉ ở trạng thái public

__call() & __callstatic()

- Khi một đối tượng gọi một phương thức, mà phương thức này chưa được định nghĩa trong class, thì phương thức `__call()` được gọi để thay thế
- Khi một đối tượng gọi một phương thức static, mà phương thức này chưa được định nghĩa trong class, thì phương thức `__callstatic()` được gọi để thay thế
- `__call()` và `__callstatic()` phải được khai báo ở trạng thái public

Lambda Functions & Closures

- Lambda là một hàm ẩn danh (hàm không có tên), chỉ đơn giản là các hàm sử dụng một lần, có thể được định nghĩa vào bất cứ lúc nào, và thường được gắn với một biến
- Closure tương tự như Lambda, ngoài ra nó có thể truy cập các biến bên ngoài phạm vi của nó

Lambda Functions & Closures

Lambda

- Anonymous functions hàm ẩn danh (hàm không có tên)
- Trong lập trình chức năng Lambda được đánh giá khá cao, bởi vì:
 - Truyền như một tham số vào hàm khác
 - Được gán cho biến
 - Được tạo và sử dụng trong thời gian thực thi

Lambda Functions & Closures

Closure

- Closure tương tự như Lambda, ngoài ra nó có thể truy cập các biến bên ngoài phạm vi của nó

Design Pattern

Thông tin chung

- Design pattern cung cấp các "mẫu thiết kế", các giải pháp để giải quyết các vấn đề chung, thường gặp trong lập trình.
- Có khá nhiều các mẫu design pattern, mỗi loại có thể được dùng để giải quyết một vấn đề nào đó, tuy nhiên đặc điểm chung là tăng tính dễ bảo trì của phần mềm

Design Pattern

Phân nhóm

- Creational Pattern: Abstract Factory, Factory Method, Singleton, Builder, Prototype → sử dụng một số thủ thuật để khởi tạo đối tượng mà không nhìn thấy từ khóa new
- Structural Pattern: Adapter, Bridge, Composite, Decorator, Facade, Proxy và Flyweight → thiết lập, định nghĩa quan hệ giữa các đối tượng.
- Behavioral Pattern: Interpreter, Template Method, Chain of Responsibility, Command, Iterator, Mediator, Memento, Observer, State, Strategy và Visitor. → thực hiện các hành vi của đối tượng.

Design Pattern

Singleton

- **What ?** Đảm bảo chỉ duy nhất một thực thể của một class được tạo ra
- **When?** Khi cần tạo ra một class mà class đó chỉ có một đối tượng duy nhất và có thể truy cập đến đối tượng đó ở bất kỳ nơi đâu

Design Pattern

Decorator

- Decorator thuộc nhóm Structural pattern
- Decorator pattern cho phép thêm method hoặc property của 1 đối tượng trong lúc chương trình thực thi

Design Pattern

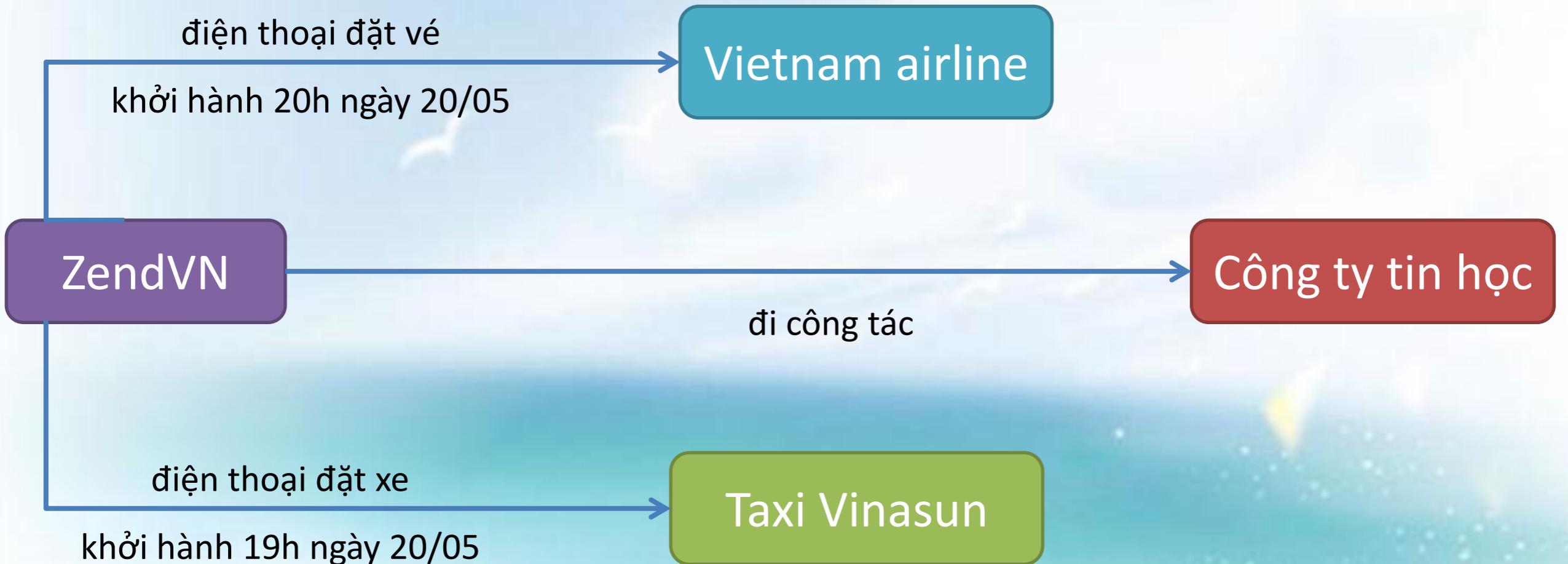
Factory

- Tạo một đối tượng mà không cần thiết chỉ ra một cách chính xác lớp nào sẽ sẽ được sử dụng để tạo đối tượng đó.
- Giảm bớt sự phụ thuộc của việc tạo đối tượng với logic của ứng dụng
- Tăng tính dễ đọc và bảo trì của ứng dụng

Dependency Injection

- Dependency Injection (DI) giúp giảm sự phụ thuộc giữa các lớp, tạo ra sự linh hoạt cao và tính dễ bảo trì cho ứng dụng
- Lợi ích khi sử dụng DI: giảm sự phụ thuộc giữa các class, tăng khả năng dùng lại và khả năng dễ đọc của mã nguồn, tạo cho ứng dụng tính mềm dẻo và khả năng nâng cấp bảo trì trở nên đơn giản và nhanh chóng hơn

Tình huống thực tế “Đi công tác”



Tình huống thực tế “Đi công tác”

- Thay đổi hãng Taxi
- Thay đổi hãng máy bay
- Thay đổi hình thức đặt vé

Tình huống thực tế “Đi công tác”



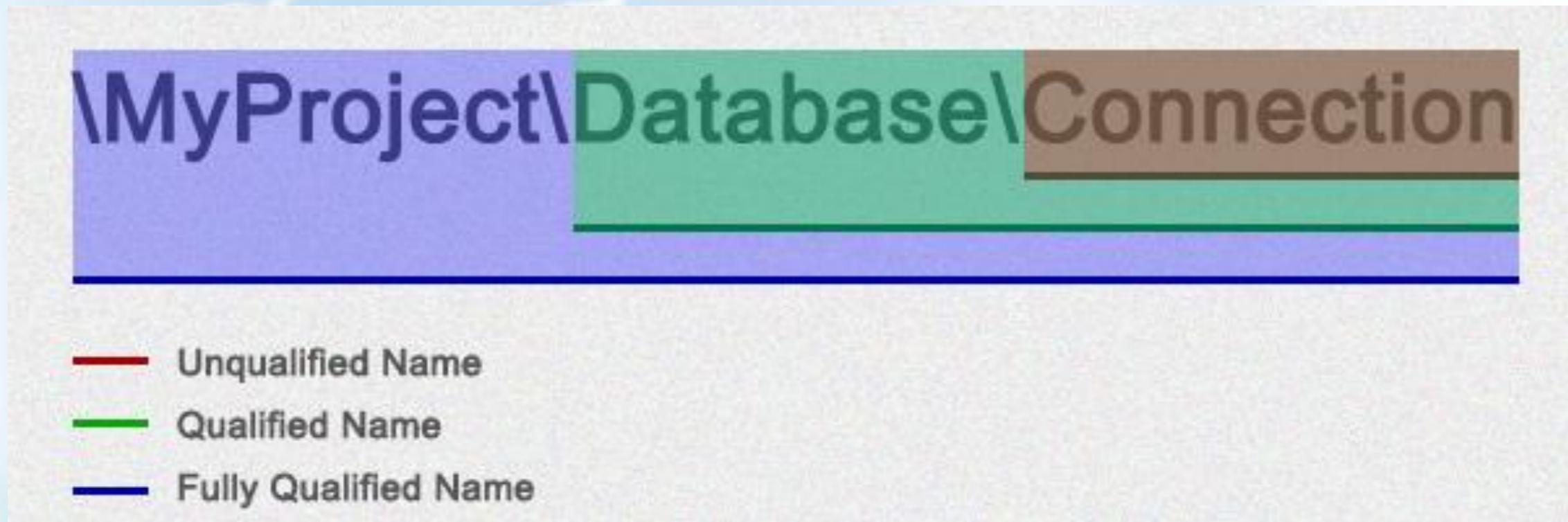
Sử dụng namespace trong PHP

- Tránh xung đột đối với tên lớp, tên hàm và tên hằng số giữa tập tin khác nhau
- Làm cho mã nguồn ngắn gọn và dễ đọc hơn

- Namespaces đơn
- Namespaces phân cấp
- Multi namespace

- Sử dụng hằng số __NAMESPACE__
- Truy xuất class, function, constant khi sử dụng namespace
- Sử dụng từ khoá use đặt alias cho namespace

Lưu ý sử dụng namespace trong PHP



Zend Skeleton Application

(tiếp xúc các kiến thức căn bản về Zend Framework thông qua việc cài đặt và tìm hiểu Zend Skeleton)

Zend Skeleton Application

Tổng quan Zend Framework

PHP Framework

Ngôn ngữ lập trình PHP

- PHP là một ngôn ngữ lập trình miễn phí được phát triển từ khá lâu và có một cộng đồng sử dụng rộng rãi.
- Với tính “nhanh – dễ – nhiều” cộng với tư duy hướng đối tượng hay mô hình MVC, ... được áp dụng trong PHP không quá khó khăn đã làm cho ngôn ngữ này có một vài trò đặc biệt quan trọng trong việc xây dựng và phát triển các ứng dụng website
- Với các ứng dụng lớn việc quá thoải mái với PHP sẽ làm hạn chế sự phát triển ? → Khái niệm PHP Framework ra đời

PHP Framework

PHP Framework

- Framework có thể hiểu là 1 bộ khung, 1 library (thư viện), 1 bộ công cụ, 1 kiến trúc phần mềm hay 1 tiêu chuẩn nào đó, ...
- PHP Framework là một framework được xây dựng và phát triển trên nền tảng ngôn ngữ PHP, giúp việc xây dựng các ứng dụng web trở nên dễ dàng, đơn giản và chuyên nghiệp hơn.

PHP Framework

Hiểu Framework như một “Tiêu chuẩn”

- Với việc áp dụng đúng các “chuẩn” của framework đặt ra sẽ giúp tăng tính bảo trì của ứng dụng, tiết kiệm thời gian phát triển, kế thừa triệt để các tinh hoa của framework.
- Một số “chuẩn” thường gặp: quy định về khai báo biến, function, class; qui định về cách truyền dữ liệu; luồng xử lý của ứng dụng; cấu trúc hệ thống tập tin và thư mục

PHP Framework

Hiểu Framework như một “Thư viện”

- Cung cấp các công cụ (function, class, ...) giải quyết các yêu cầu thường gặp khi xây dựng ứng dụng: thao tác database, thao tác file, email, phân quyền, ...
- Tiết kiệm thời gian xây dựng và đỡ gây nhàm chán khi phải xây dựng các mã nguồn được sử dụng lặp đi lặp lại

PHP Framework

Hiểu Framework như một “Kiến trúc phần mềm”

- Vấn đề xây dựng ứng dụng trên PHP thuần: cách tổ chức cấu trúc thư mục, cách phân chia chức năng cho các team, áp dụng mô hình MVC, design pattern, ... ?
- Bản thân một framework đã được tích hợp sẵn các thành phần trên, chúng ta chỉ cần tuân thủ các nguyên tắc của nó để đảm bảo việc xây dựng một ứng dụng chuyên nghiệp

PHP Framework

Ưu điểm khi sử dụng PHP Framework để phát triển

- Tiết kiệm thời gian trong xây dựng và phát triển
- Xây dựng ứng dụng chuyên nghiệp và kế thừa các ưu điểm của framework đó
- Hiệu năng cũng như bảo mật sẽ được chăm sóc và cập nhật thường xuyên nhờ cộng đồng phát triển framework đó.
- Phát triển các chức năng thông qua tích hợp các module, extension, plugin vào ứng dụng mà không tốn sức để phát triển các chức năng đó

PHP Framework

Các PHP Framework phổ biến

- Zend Framework
- CakePHP
- CodeIgniter
- Symfony

Zend Framework

- Zend Framework (ZF) là một PHP Framework được định hướng theo mô hình MVC và là một PHP framework ra đời khá trễ, chính vì vậy ZF đã thừa hưởng những tinh hoa của các framework khác và tránh khỏi những sai lầm của các framework trước đã mắc phải trong quá trình phát triển.
- ZF có tính mềm dẻo, linh hoạt và dễ dàng tích hợp các thư viện PHP có sẵn.
- ZF thường được ứng dụng vào các dự án Open Source và các công ty lớn
- Website <http://framework.zend.com/>

Zend Skeleton Application

Download và Cài đặt

- Download & Cài đặt
- Import project vào Zend Studio
- Kiểm tra kết quả

Zend Skeleton Application

Tìm hiểu cấu trúc ứng dụng

Zend Skeleton Application

Tìm hiểu cấu trúc thư mục

Tập tin / Thư mục	Thông tin / Chức năng
config	Folder chứa các cấu hình
data	Folder chứa các tài file dữ liệu được tạo bởi ứng dụng
module	Folder chứa các module
public	Folder chứa các tập tin thư mục dùng chung
vendor	Folder chứa thư viện

Zend Skeleton Application

Tìm hiểu cấu trúc module

Tập tin / Thư mục	Thông tin / Chức năng
config	Folder chứa các cấu hình cho module
language	Folder chứa các tài nguyên ngôn ngữ
src/Application	Folder chứa các controller và model của module hiện tại
view/application	Folder chứa view cho các controller
view/error	Folder chứa view cho các thông báo lỗi
view/layout	Folder chứa các layout

Zend\Config

- Lớp Zend\Config được xây dựng để làm việc và thao tác với các file cấu hình, các file cấu hình có thể ở định dạng ini, xml, json hoặc thậm chí là một mảng cấu hình của PHP
- Tìm hiểu các lớp
 - Zend\Config\Config
 - Zend\Config\Processor
 - Zend\Config\Reader
 - Zend\Config\Writer

Tổng quan về Luồng cấu hình trong ZF 2

System configuration

config/application.config.php

Cấu hình được chuyển đến trình ứng dụng, và sau đó là Module manager để kích hoạt hệ thống

Được sử dụng ngay lập tức, không kết hợp và không bị ghi đè với bất kỳ cấu hình nào khác

ConfigListener kết hợp các cấu hình và gửi về ServiceManager

Application configuration

ModuleManager đọc các cấu hình của các Module

Thứ tự đọc là thứ tự được khai báo trong System Configuration

Kết hợp các phần tử service_manager ở mỗi module

Kết hợp cấu hình được trả bởi Module::getConfig()

Tập tin cấu hình được thiết lập ở config_glob_paths được kết hợp lại dựa trên thứ tự được khai báo

Module Configuration

Module cung cấp cấu hình riêng của nó cho ứng dụng, có 2 cơ chế cho việc này

- Implement Zend\ModuleManager\Feature\ConfigProviderInterface thông qua getConfig trả về thông tin cấu hình của module (ModuleManager tự động gọi getConfig và getAutoloaderConfig)
- Implement từ khá nhiều các class và interface khác để trả về thông tin cấu hình

Configuration mapping table

Configuration mapping table

Manager name	Interface name	Module Method name	Config key name
ControllerPluginManager	ControllerPluginProviderInterface	getControllerPluginConfig()	controller_plugins
ControllerManager	ControllerProviderInterface	getControllerConfig()	controllers
FilterManager	FilterProviderInterface	getFilterConfig()	filters
FormElementManager	FormElementProviderInterface	getFormElementConfig()	form_elements
HydratorManager	HydratorProviderInterface	getHydratorConfig()	hydrators
InputFilterManager	InputFilterProviderInterface	getInputFilterConfig()	input_filters
RoutePluginManager	RouteProviderInterface	getRouteConfig()	route_manager
SerializerAdapterManager	SerializerProviderInterface	getSerializerConfig()	serializers
ServiceLocator	ServiceProviderInterface	getServiceConfig()	service_manager
ValidatorManager	ValidatorProviderInterface	getValidatorConfig()	validators
ViewHelperManager	ViewHelperProviderInterface	getViewHelperConfig()	view_helpers
LogProcessorManager	LogProcessorProviderInterface	getLogProcessorConfig	log_processors
LogWriterManager	LogWriterProviderInterface	getLogWriterConfig	log_writers

Autoloader

- Autoloading là quá trình tự động nạp các lớp khi có yêu cầu sử dụng từ lớp đó
 - Standart Autoloader
 - Classmap Autoloader

Xử lý dữ liệu

(Sử dụng các lớp Filter, Serializer, Feed, Dom, Escaper ... để lọc, phân tích và chuyển đổi dữ liệu)

Xử lý dữ liệu

Zend\Filter

Zend\Filter

Giới thiệu

- Zend\Filter lọc, phân tích và truy xuất các nội dung trong một chuỗi nào đó
- Các nội dung cần tìm hiểu
 - Các lớp Filter căn bản
 - Word Filter
 - Filter Chains
 - Writing Filter

Các lớp Filter căn bản (1)

Filter Class	Input	Output
Zend\I18n\Filter\Alnum	abc !@#\$^&*(dec 12 ()*() 12	abcdef1212
Zend\I18n\Filter\Alpha	abc !@#\$^&*(dec 12 ()*() 12	abcdec
Zend\Filter\BaseName	www zend vn/public/test.html	test.html
Zend\Filter\Dir	www zend vn/public/test.html	www zend vn/public/
Zend\Filter\Callback	Gọi các phương thức xử lý khác đã được xây dựng sẵn	
Zend\Filter\Digits	Zend Framework 2.34 !@#	234
Zend\Filter\StringToLower	Khóa học Zend Framework 2	khóa học zend framework 2
Zend\Filter\StringToUpper	Khóa học Zend Framework 2	KHÓA HỌC ZEND FRAMEWORK 2

Các lớp Filter căn bản (2)

Filter Class	Input	Output
Zend\Filter\StringTrim	Zend Framework 2	Zend Framework 2
Zend\Filter\StripTags	<h3>Khóa học lập trình Zend Framework 2</h3>	Khóa học lập trình Zend Framework 2
Zend\Filter\Compress	Abc.txt	Abc.zip
Zend\Filter\Decompress	Abc.zip	Abc.txt
Zend\Filter\PregReplace	Sử dụng biểu thức chính quy trong Filter	
Zend\Filter\HtmlEntities	This is a test " <>	This is a test " <>

Word Filter (1)

Filter Class	Input	Output
Zend\Filter\Word\CamelCaseToDash	AbcDefGeh	Abc-Def-Geh
Zend\Filter\Word\CamelCaseToSeparator	AbcDefGeh	Abc(x)Def(x)Geh
Zend\Filter\Word\CamelCaseToUnderscore	AbcDefGeh	Abc_Def_Geh
Zend\Filter\Word\DashToCamelCase	Abc-Def-Geh	AbcDefGeh
Zend\Filter\Word\DashToSeparator	Abc-Def-Geh	Abc(x)Def(x)Geh
Zend\Filter\Word\DashToUnderscore	Abc-Def-Geh	Abc_Def_Geh

Word Filter (2)

Filter Class	Input	Output
Zend\Filter\Word\SeparatorToCamelCase	Abc(x)Def(x)Geh	AbcDefGeh
Zend\Filter\Word\SeparatorToDash	Abc(x)Def(x)Geh	Abc-Def-Geh
Zend\Filter\Word\SeparatorToSeparator	Abc(x)Def(x)Geh	Abc(x)Def(x)Geh
Zend\Filter\Word\UnderscoreToCamelCase	Abc_Def_Geh	AbcDefGeh
Zend\Filter\Word\UnderscoreToSeparator	Abc_Def_Geh	Abc(x)Def(x)Geh
Zend\Filter\Word\UnderscoreToDash	Abc_Def_Geh	Abc-Def-Geh

Xử lý dữ liệu

Zend\Serializer

Zend\Serializer

- Zend\Serializer chuyển đổi các đối tượng PHP thành các chuỗi có thể lưu trữ được.
- Các nội dung cần tìm hiểu
 - Các gọi thông thường hoặc thông qua Factory
 - serialize()
 - unserialize()

Xử lý dữ liệu

Zend\Escaper

Zend\Escaper

- Zend\Escaper cung cấp bộ lọc trong quá trình xuất dữ liệu (data output), đối với các dữ liệu đầu vào nên sử dụng Zend\Filter, HTML Purifier để lọc dữ liệu
- Các phương thức cần tìm hiểu
 - escapeHtml *xuất nội dung HTML*
 - escapeHtmlAttr *xuất nội dung thuộc tính của HTML*
 - escapeJs *xuất nội dung JS*
 - escapeCss *xuất nội dung CSS*
 - escapeUrl *xuất nội dung URL*

XSS (Cross-Site Scripting)

- XSS (Cross-Site Scripting) kỹ thuật tấn công website phổ biến hiện nay bằng cách thông qua các form nhập liệu chèn vào những thẻ HTML hoặc những đoạn mã script nguy hiểm, độc hại để tấn công website
- Các tác hại:
 - Xâm phạm cookies, mật khẩu, session làm việc
 - Cài đặt các loại virus, trojan, backdoor trên máy tính client
 - Tuy không thay đổi được cấu trúc mã nguồn, cơ sở dữ liệu nhưng nó tấn công vào bìa mặt của website

Xử lý dữ liệu

HTML Purifier

HTML Purifier

- HTML Purifier là một thư viện miễn phí được xây dựng trên ngôn ngữ lập trình PHP. Thư viện này được sử dụng để phòng chống các tấn công bằng XSS và làm cho dữ liệu HTML được “sạch”
- Website: <http://htmlpurifier.org/>
- Các nội dung cần tìm hiểu
 - Điểm qua một số config của HTML Purifier
 - Tích hợp HTML Purifier thành một filter của Zend Framework

Xử lý dữ liệu

Zend\Dom

Zend\Dom

- Zend\Dom được xây dựng nhằm truy xuất đến các tài liệu có cấu trúc được đơn giản và dễ dàng hơn
- Zend\Dom\Query sử dụng Xpath và CSS Selector cho các truy vấn XML và (X)HTML
- Các nội dung cần tìm hiểu
 - execute(\$query): query the document using CSS selector notation.
 - queryXpath(\$xPathQuery): query the document using XPath notation.

Xử lý dữ liệu

Zend\Json

Zend\JSON

- JSON (JavaScript Object Notation) được sử dụng cho việc trao đổi dữ liệu giữa JavaScript và các ngôn ngữ khác
- Zend\JSON chuyển đổi các đối tượng, các mảng trong PHP thành các chuỗi JSON
- Các phương thức cần tìm hiểu
 - encode: chuyển đổi một đối tượng PHP sang một chuỗi JSON
 - decode: chuyển đổi một chuỗi Json sang một đối tượng PHP
 - prettyPrint: in chuỗi JSON

Zend MVC

(tìm hiểu các thành phần Event, Service, Controller, View, ...)

Zend MVC

EventManager

EventManager

Giới thiệu

- EventManager là một thành phần quan trọng trong hệ thống MVC.
- EventManager cho phép lập trình hướng sự kiện (một lớp có thể công bố sự kiện cho các đối tượng khác lắng nghe, và sau đó đưa ra các xử lý phù hợp khi sự kiện đó được kích hoạt)
- EventManager chứa danh sách các listener đối với một hoặc nhiều event
- Các từ khóa EventManager, listener, callback, event, SharedEventManager, attach, detach, trigger

EventManager

Liên tưởng

EventManager phân công công việc (*attach*) cho các nhân viên (*listener*) để tiến hành tổ chức (*trigger*) các sự kiện (*event*). Mỗi sự kiện (*event*) có thể có một hoặc nhiều công việc (*callback*) cần phải thực hiện

- Một nhân viên (*listener*) có thể nhận một hoặc nhiều sự kiện (*event*)
- Một sự kiện (*event*) có thể do một hoặc nhiều nhân viên (*listener*) phụ trách
- Một sự kiện (*event*) có thể có một hoặc nhiều công việc (*callback*) cần phải thực hiện
- Đối với một sự kiện (*event*) các công việc (*callback*) có độ ưu tiên khác nhau (*priority*)

Zend MVC

ServiceManager

ServiceManager

Giới thiệu

- ServiceManager (SM) và Event Manager (EM) là 2 thành phần khá quan trọng và linh hoạt trong các ứng dụng Zend Framework 2
 - SM đảm bảo tất cả các thành phần ở trạng thái có sẵn và sẵn sàng được sử dụng trên ứng dụng
- Tận dụng được các ưu điểm của SM nếu chúng ta đầu tư tốt trong việc thiết kế các lớp và biết rõ những công việc chúng ta cần thực hiện

ServiceManager

Cấu hình

- Việc cấu hình SM thường đặt ở các *tập tin cấu hình* và phương thức `getServiceConfig()` được khai báo ở tập tin `Module.php`

```
80 return array(
81     'service_manager' => array(
82         'abstract_factories'=> array(),
83         'aliases'           => array(),
84         'factories'         => array(),
85         'initializers'      => array(),
86         'invokables'        => array(),
87         'services'          => array(),
88         'shared'            => array(),
89     ),
90 );
```

View Helper

View Helper tập hợp các lớp được implements từ Zend\View\Helper\HelperInterface, được dùng ở ngoài View trong các trường hợp định dạng, tạo phần tử, xuất title, css, js, ...

- | | | |
|-------------------|-------------------------|-----------------|
| 1. BasePath | 8. HeadScript | 15. JSON |
| 2. Cycle | 9. HeadStyle | 16. Partial |
| 3. Doctype | 10. HeadTitle | 17. Placeholder |
| 4. FlashMessenger | 11. HtmlList | 18. Url |
| 5. Gravatar | 12. HTML Object Plugins | 19.... |
| 6. HeadLink | 13. Identity | |
| 7. HeadMeta | 14. InlineScript | |

View Helper

Thay đổi layout đang hiển thị

- Tại sao xuất hiện giao diện của ứng dụng như vậy
- Cách thay đổi layout đang hiển thị

View Helper

Zend\View\Helper\BasePath

- Helper for retrieving the base path

```
<a href="<?php echo $this->basePath(); ?>">basePath</a>

<link rel="stylesheet" type="text/css"
      href="<?php echo $this->basePath('css/base.css'); ?>">
/>
```

View Helper

Zend\View\Helper\Cycle

- Helper for alternating between set of values

```
<table>
    <?php foreach ($this->course as $book): ?>
        <tr style="background-color:
                    <?php echo $this->cycle(array('red', 'yellow'))->next() ?>">
            <td><?php echo $this->escapeHtml($book['name']) ?></td>
        </tr>
    <?php endforeach ?>
</table>
```

View Helper

Zend\View\Helper\Doctype

- Helper for setting and retrieving the doctype
- Xem tài liệu hiện tại và kiểm tra phần doctype ? → bổ sung trong layout

```
<?php echo $this->docType(); ?>
```

View Helper

Zend\View\Helper\EscapeX

- Zend\View\Helper\EscapeCss
- Zend\View\Helper\EscapeHtml
- Zend\View\Helper\EscapeAttr
- Zend\View\Helper\EscapeJs
- Zend\View\Helper\EscapeUrl

```
<?php echo $this->escapeHtml($value); ?>
```

View Helper

Zend\View\Helper\Gravatar

- Helper for retrieving avatars from gravatar.com
- Methods: setEmail(), setImgSize(), setDefaultImg(), setSecure()
- More: <http://localhost/zend2-api/class-Zend.View.Helper.Gravatar.html>

```
$settings = array(  
    'img_size'      => 40,  
    'default_img'   => \Zend\View\Helper\Gravatar::DEFAULT_MM,  
    'secure'        => null,  
);  
$email = 'email@example.com';  
echo $this->gravATAR($email, $settings);
```

View Helper

Zend\View\Helper\HeadTitle

- Helper for setting and retrieving title element for HTML head.
- Thiết lập ở Module.php
- Thiết lập ở Controller
- Thiết lập ở View

View Helper

Zend\View\Helper\HeadMeta

- Meta Name: appendName(), offsetSetName(), setName()
- Meta HttpEquiv: appendHttp_equiv(), offsetSetHttp_equiv(), setHttp_equiv()
- Charset: setCharset(\$charset)
- Clear Meta: getContainer()->exchangeArray(array())

View Helper

Zend\View\Helper\HeadLink

```
$renderer = $this->getServiceLocator()->get('Zend\View\Renderer\PhpRenderer');
$headLink = $renderer->headLink();

$headLink->appendStylesheet(PUBLIC_URL . '/css/style.css', 'screen');
$headLink->appendStylesheet(PUBLIC_URL . '/css/bootstrap.css', 'screen');
$headLink->appendStylesheet(PUBLIC_URL . '/css/bootstrap1.css', 'screen', 'lte IE6');

$headLink->offsetSetStylesheet(0, PUBLIC_URL . '/css/bootstrap1.css', 'screen');
$headLink->getContainer()->exchangeArray(array());
```

View Helper

Zend\View\Helper\HeadScript

```
$renderer    = $this->getServiceLocator()->get('Zend\View\Renderer\PhpRenderer');
$headScript = $renderer->headScript();

$headScript->appendFile(PUBLIC_URL . '/js/jquery.min.js', 'text/javascript');
$headScript->appendFile(PUBLIC_URL . '/js/bootstrap.js', 'text/javascript',
array('conditional' => 'lt IE 7'));
$headScript->offsetSetFile(0, PUBLIC_URL . '/js/bootstrap.min.js', 'text/javascript');
```

View Helper

Zend\View\Helper\InlineScript

- `InlineScript` should be used when you wish to include scripts inline in the HTML body

```
$viewhelpermanager      = $this->getServiceLocator()->get('viewhelpermanager');
$script                 = $viewhelpermanager->get('inlineScript');
$script->appendScript(
    '$(document).ready(function() { alert("Hello"); });',
    'text/javascript',
    array('noescape' => true)); // Disable CDATA comments
```

View Helper

Zend\View\Helper\HtmlList

- Helper for ordered and unordered lists

```
$this->htmlList($items, $ordered, $attribs, $escapes);
```

View Helper

Zend\View\Helper\HtmlFlash

- The HtmlFlash is used for embedding media like Flash or QuickTime in web pages

```
echo $this->htmlFlash($data, $attribs, $params, $content);
```

View Helper

Zend\View\Helper\Json

- Helper for simplifying JSON responses

```
echo $this->json($arrayData);
```

View Helper

Zend\View\Helper\Partial

- Helper for rendering a template fragment in its own variable scope

```
<?php
echo $this->partial('course.phtml', array(
    'name' => 'Khóa học lập trình PHP',
    'description' => 'PHP đã và đang trở thành một ngôn ngữ lập trình phổ...',
    'attributes' => array(
        'color' => 'yellow',
        'font-weight'=> 'bold'
    )
));
?>
```

View Helper

Zend\View\Helper\PartialLoop

- Helper for rendering a template fragment in its own variable scope; iterates over data provided and renders for each iteration.

```
echo $this->partialLoop('course.phtml', $arrCourse);
```

View Helper

Zend\View\Helper\Placeholder

```
$this->placeholder('fooTwo')->exchangeArray($arrCourse);
$this->placeholder('fooTwo')->setPrefix('<ul><li>')
    ->setPostfix('</li></ul>')
    ->setSeparator('</li><li>');

echo $this->placeholder('fooTwo');
```

```
<?php $this->placeholder('foo')->captureStart(); ?>
<div>
    <h3 style="color:blue">Khóa học Lập trình PHP</h3>
    <p>PHP ngôn ngữ lập trình phổ biến trong các ứng dụng web... </p>
</div>
<?php $this->placeholder('foo')->captureEnd() ?>

<?php echo $this->placeholder('foo') ?>
```

View Helper

\$this->currencyFormat()

```
echo $this->currencyFormat (float $number, string $currencyCode, bool  
$showDecimals, string $locale)
```

- \$number – The value to format
- \$currencyCode – mã code được sử dụng để định dạng tiền tệ
- \$showDecimals – hiển thị số thập phân ?
- \$locale – mã vùng được sử dụng để định dạng tiền tệ

View Helper

\$this->dateFormat()

```
echo $this->dateFormat ($date, int $dateType, int $timeType, string $locale)
```

- \$date – The value to format (DateTime object)
- \$dateType – none, short, medium, long, full (*IntlDateFormatter::NONE*)
- \$timeType – none, short, medium, long, full (*IntlDateFormatter::NONE*)
- \$locale – Locale in which the date would be formatted

View Helper

\$this->numberFormat()

```
echo $this->numberFormat ($number, int $formatStyle, int $formatType, string  
$locale)
```

- \$number – The value to format
- \$formatStyle – Style of the formatting
- \$formatType – The formatting type to use
- \$locale – Locale in which the number would be formatted

View Helper

Zend\View\Helper\ServerUrl

- Helper for returning the current server URL
- Methods `getHost()`, `getPort()`, `getScheme()`

```
$serverURL      =      $this->serverUrl();
echo '<h3 style="color:red;">$serverURL: ' . $serverURL . '</h3>';
```

View Helper

- View Helper – FlashMessenger
- View Helper – Identity
- View Helper – URL
- ...

Zend View

Tìm hiểu các key word trong view_manager

```
'view_manager' => array(
    'display_not_found_reason' => true,
    'not_found_template' => 'error/404',
    'display_exceptions' => true,
    'exception_template' => 'error/index',
    'doctype' => 'HTML5',
    'template_map' => array(
        'layout/layout' => __DIR__ . '/../view/layout/layout.phtml',
        'error/404' => __DIR__ . '/../view/error/404.phtml',
        'error/index' => __DIR__ . '/../view/error/index.phtml',
    ),
    'template_path_stack' => array( __DIR__ . '/../view'),
    'default_template_suffix' => 'php',
    'layout' => 'layout/layout'
),
```

Zend View

Truyền dữ liệu từ Controller ra View

```
return array(
    'message' => 'Hello world',
    'message2' => 'Hello world 2',
);
```

```
$view = new ViewModel();
$view->setVariables(array(
    'variable_one' => 'Some Variable',
    'variable_two' => 'Some other Variable',
));
return $view;
```

Zend Router

Lý thuyết

- Định nghĩa các quy tắc ánh xạ, để từ các request từ phía người dùng (URL) và dựa vào các quy tắc ánh xạ đã đăng ký xác định cần phải thực hiện module nào, controller nào và action nào
- zend2.com/register.html → zend2.com/public/home/user/register

Zend Router

Tìm hiểu các lớp router

- Zend\Mvc\Router\Http\Hostname
- Zend\Mvc\Router\Http\Literal
- Zend\Mvc\Router\Http\Part
- Zend\Mvc\Router\Http\Regex
- Zend\Mvc\Router\Http\Scheme
- Zend\Mvc\Router\Http\Segment
- Zend\Mvc\Router\Http\Wildcard
- Zend\Mvc\Router\Http\Query
- Zend\Mvc\Router\Http\Method

Zend Router

Chuẩn bị

- Xóa public trong URL hiện tại
- Tạo virtual host
- Tách phần cấu hình router trong file module.config.php
- Nhận tham số được truyền

Zend Router

Zend\Mvc\Router\Http\Hostname

```
$hostnameRoute = array(
    'type' => 'hostname',
    'options' => array(
        'route' => 'zendskeleton.training.com',
        'defaults' => array(
            '__NAMESPACE__' => 'Mvc\Controller',
            'controller' => 'Index',
            'action' => 'index'
        )
    )
);
```

Zend Router

Zend\Mvc\Router\Http\Literal

```
$literalRoute = array(
    'type'      => 'Literal',
    'options'   => array(
        'route'    => '/contact',
        'defaults' => array(
            '__NAMESPACE__' => 'Mvc\Controller',
            'controller'     => 'Index',
            'action'         => 'index'
        ),
    ),
);
```

Zend Router

Zend\Mvc\Router\Http\Scheme

```
$schemeAccountRoute = array(
    'type'          => 'Scheme',
    'options'       => array(
        'route'        => '/account',
        'scheme'       => 'http',
        'defaults'     => array(
            '__NAMESPACE__' => 'Mvc\Controller',
            'controller'   => 'Index',
            'action'        => 'account',
        ),
    ),
);
```

Zend Router

Zend\Mvc\Router\Http\Segment

```
$segmentRoute      = array(
    'type' => 'Segment',
    'options' => array(
        'route' => '/:module[/]',
        'constraints' => array(
            'module'          => 'm..',
        ),
        'defaults' => array(
            '__NAMESPACE__'  => 'Mvc\Controller',
            'controller'     => 'index',
            'action'         => 'segment',
        ),
    ),
);
```

Zend Router

Zend\Mvc\Router\Http\Regex

```
$regexRoute = array(
    'type'          => 'Regex',
    'options'       => array(
        'regex'         => '/course/(?<name>[a-zA-Z0-9-]+)-(?<id>[0-
                                9]+)\.(?<extension>(html|php|asp))',
        'defaults'     => array(
            '__NAMESPACE__'  => 'Mvc\Controller',
            'controller'    => 'Router',
            'action'        => 'course',
            'name'          => 'zend-2x',
            'id'            => '100',
            'extension'     => 'phtml',
        ),
        'spec'          => '/course/%name%-%id%.%extension%',
    ),
);
```

Zend Router

Router Types

- Simple Routing
- Tree Routing
 - Kết hợp các router lại với nhau
 - Một router bất kỳ có thể có nhiều router con, các router con chỉ được kích hoạt khi router cha được khớp với URI
 - Các router con thuộc về các dạng Simple Routing

Zend Router

Tree Routing

```
$route = array(  
    'type'      => 'Literal',  
    'options'   => array(  
        'route'          => '/account',  
        'defaults'       => array(),  
    ),  
    'may_terminate'=> true,  
    'child_routes'=> array(  
        'query' => array(  
            'type' => 'segment',  
            'options' => array(  
                'route' => '/profile[/type/:upload]',  
                'defaults' => array( 'action' => 'course' ),  
                'constraints' => array( 'upload'=> '[0,1]' ),  
            ),  
        ),  
    ),  
);
```

Zend Controller

- Controller điều hướng và xử lý các request , trả kết quả cho View để hiển thị dữ liệu cho người dùng
- Các bước để tạo controller
 - Tạo file và class
 - Đăng ký trong tập tin module
 - Đăng ký router (nếu cần thiết)

Controller Plugins

Tìm hiểu các controller plugins (Zend\Mvc\Controller\Plugin\)

- AcceptableViewModelSelector
- FlashMessenger
- Forward
- Identity
- Layout
- Params
- PostRedirectGet
- Redirect
- Url

Controller Plugins

Zend\Mvc\Controller\Plugin\Forward

```
public function forwardAction(){
    return $this->forward()->dispatch('Mvc/controller/Index', array(
        'action' => 'detail',
        'params' => array(
            'name'=> 'John smith',
            'age'=> 21,
        )
    ));
}

public function detailAction(){
    $object = $this->params('params');
    return false;
}
```

Controller Plugins

Zend\Mvc\Controller\Plugin\Layout

```
$this->layout()->setTemplate('view/abc');

$this->layout('layout/home');
```

Controller Plugins

Zend\Mvc\Controller\Plugin\Params

Params Plugin giúp nhận các tham số ở Action từ các nguồn khác nhau

```
$paramPost= $this->params()->fromPost();  
  
echo $paramQuery= $this->params()->fromQuery('username');  
  
$paramRoute= $this->params()->fromRoute('name');  
  
$paramHeader= $this->params()->fromHeader();  
  
$paramFiles= $this->params()->fromFiles();
```

Controller Plugins

Zend\Mvc\Controller\Plugin\Redirect

Redirect (sự chuyển hướng) là quá trình chuyển từ URL này đến một URL khác

```
$this->redirect()->toUrl('http://www.zend.vn');

$this->redirect()->toRoute('regexRoute', array(
    'name'=> 'lap-trinh-zend-framework-2x',
    'id'=> 100,
    'extension' => 'html'
));
```

Controller Plugins

Zend\Mvc\Controller\Plugin\Url

```
$params= array(
    'name'          => 'lap-trinh-ios',
    'id'            => 989,
    'extension'     => 'php'
);

$router = $this->getEvent()->getRouter();
$url   = $router->assemble($params, array('name' => 'regexRoute'));

$url = $this->url()->fromRoute('regexRoute', $params);
```

Controller Plugins

Zend\Mvc\Controller\Plugin\FlashMessenger (P1)

```
public function flashMessenger01Action() {  
  
    $flashMessenger = $this->flashMessenger();  
    $flashMessenger->addInfoMessage('InfoMessage 1');  
    $flashMessenger->addInfoMessage('InfoMessage 2');  
    if($flashMessenger->hasCurrentInfoMessages()){  
        $infoMessages = $flashMessenger->getInfoMessages();  
    }else{  
        echo '<h3 style="color:red;">No message</h3>';  
    }  
}
```

Controller Plugins

Zend\Mvc\Controller\Plugin\FlashMessenger (P2)

```
public function flashMessenger02Action(){

    $flashMessenger = $this->flashMessenger();
    $flashMessenger->addInfoMessage('InfoMessage 1');
    $flashMessenger->addInfoMessage('InfoMessage 2');
    $flashMessenger->addErrorMessages('ErrorMessage 1');

    $infoMessages      = $flashMessenger->getInfoMessages();
    $errorMessages     = $flashMessenger->getErrorMessages();

    $flashMessenger->clearMessagesFromNamespace('info');
    $flashMessenger->clearMessagesFromContainer();

    return false;
}
```

Controller Plugins

Zend\Mvc\Controller\Plugin\FlashMessenger (P3)

```
public function flashMessenger03Action(){
    $flashMessenger=  $this->flashMessenger();

    $namespace= $flashMessenger->getNamespace();
    $flashMessenger->setNamespace('zendvn');
    $flashMessenger->addMessage('Zend framework 2.x 1');

    if($flashMessenger->hasMessages()){
        $messages = $flashMessenger->getMessages();
    }else{
        echo '<h3 style="color:red;">No message</h3>';
    }

    return false;
}
```

Controller Plugins

Zend\Mvc\Controller\Plugin\FlashMessenger (P4)

```
public function flashMessenger05Action(){
    $this->flashMessenger()->addSuccessMessage('Bạn đã đăng nhập thành công!');
    return $this->redirect()->toRoute('mvcRoute/default', array(
        'controller'=> 'index',
        'action'=> 'success'
    ));
}

public function successAction(){
    $flashMessenger = $this->flashMessenger();

    if ($flashMessenger->hasSuccessMessages()) {
        $result['messages'] = $flashMessenger->getSuccessMessages();
    }
    return $result;
}
```

Controller Plugins

Zend\Mvc\Controller\Plugin\FlashMessenger (P5)

```
echo $this->flashMessenger()->render('error', array('alert', 'alert-danger'));  
  
echo $this->flashMessenger()  
    ->setMessageOpenFormat('<div><h4>')  
    ->setMessageSeparatorString('</h4><h4>')  
    ->setMessageCloseString('</h4></div>')  
    ->render('error');
```

```
'view_helper_config' => array(  
    'flashmessenger' => array(  
        'message_open_format'      => '<div><h4>',  
        'message_close_string'    => '</h4><h4>',  
        'message_separator_string' => '</h4></div>'  
    )  
)
```

Controller Plugins

Xây dựng Controller Plugin

- Tạo tập tin plugin ở thư mục src/Mvc/Controller/Plugin/MyPlugin.php
- Xây dựng nội dung tập tin
- Đăng ký plugin trong tập tin cấu hình module.config.php hoặc phương thức getControllerPluginConfig
- Sử dụng plugin vừa tạo

Controller Plugins

Zend\Mvc\Controller\Plugin\Url

```
$params= array(
    'name'          => 'lap-trinh-ios',
    'id'            => 989,
    'extension'     => 'php'
);

$router = $this->getEvent()->getRouter();
$url   = $router->assemble($params, array('name' => 'regexRoute'));

$url = $this->url()->fromRoute('regexRoute', $params);
```

Controller Plugins

Zend\Mvc\Controller\Plugin\Url

```
$params= array(
    'name'          => 'lap-trinh-ios',
    'id'            => 989,
    'extension'     => 'php'
);

$router = $this->getEvent()->getRouter();
$url    = $router->assemble($params, array('name' => 'regexRoute'));

$url = $this->url()->fromRoute('regexRoute', $params);
```

Controller Plugins

Zend\Mvc\Controller\Plugin\Url

```
$params= array(
    'name'          => 'lap-trinh-ios',
    'id'            => 989,
    'extension'     => 'php'
);

$router = $this->getEvent()->getRouter();
$url   = $router->assemble($params, array('name' => 'regexRoute'));

$url = $this->url()->fromRoute('regexRoute', $params);
```

Controller Plugins

Zend\Mvc\Controller\Plugin\Url

```
$params= array(
    'name'          => 'lap-trinh-ios',
    'id'            => 989,
    'extension'     => 'php'
);

$router = $this->getEvent()->getRouter();
$url    = $router->assemble($params, array('name' => 'regexRoute'));

$url = $this->url()->fromRoute('regexRoute', $params);
```

Controller Plugins

Zend\Mvc\Controller\Plugin\Url

```
$params= array(
    'name'          => 'lap-trinh-ios',
    'id'            => 989,
    'extension'     => 'php'
);

$router = $this->getEvent()->getRouter();
$url   = $router->assemble($params, array('name' => 'regexRoute'));

$url = $this->url()->fromRoute('regexRoute', $params);
```

FORM

(tìm hiểu các thành phần Form, Validate, cài đặt plugin, file và upload file, sử dụng Ajax trong Zend Framework, ...)

Validator

Giới thiệu

- Zend Validator được sử dụng để kiểm tra dữ liệu có phù hợp với một hoặc nhiều yêu cầu nào đó hay không.
- Tìm hiểu cách sử dụng của các lớp Zend\Validator phổ biến
 - Cách khai báo và sử dụng
 - Tùy biến thông báo lỗi
 - Kết hợp nhiều validator
- Tìm hiểu cách tự xây dựng một validator

Validator

Các lớp Validator phổ biến

- Between
- Date
- Db\RecordExists
- Db\NoRecordExists
- Digits
- EmailAddress
- File\Extension
- File\Exists
- File\Size
- File\WordCount
- File\IsImage
- File\ImageSize
- File\IsCompressed
- GreaterThan
- LessThan
- InArray
- NotEmpty
- Regex
- StringLength

Validator

Tùy biến thông báo lỗi

```
$validator->setMessages(array(
    \Zend\Validator\StringLength::TOO_SHORT =>
        "Giá trị đầu vào <b>%value%</b> phải Lớn hơn %min% ký tự",
    \Zend\Validator\StringLength::TOO_LONG =>
        "Giá trị đầu vào <b>%value%</b> phải nhỏ hơn %max% ký tự",
));
```

```
$validator = new \ZendVN\Validator\StringLength(array(
    'max' => 9,
    'min' => 7
));
```

Validator

ValidatorChain

```
$validatorChain = new \Zend\Validator\ValidatorChain();

$validatorChain->attach(new \Zend\Validator\StringLength(array(
    'min' => 6,
    'max' => 12)), true)
    ->attach(new \Zend\Validator\EmailAddress());
```

Validator

Xây dựng lớp Validate

- Lớp validate kiểm tra mã số khóa học hợp lệ
 - Cố định pattern (mẫu khóa học)
 - Truyền vào pattern
- Lớp validator kiểm tra password và confirm password hợp lệ

Zend File

Quá trình upload tập tin

- Xây dựng Form (enctype="multipart/form-data")
- Kiểm tra điều kiện upload (extension, size, width, height, ...)
- Nhận tập tin upload và xử lý
 - Upload & giữ lại tên gốc
 - Upload & đổi tên tập tin

Zend File

Thao tác upload với một tập tin

```
if($this->request->isPost()){
    $upload = new \Zend\File\Transfer\Adapter\Http();
    $fileInfo= $upload->getFileInfo();
    $fileSize= $upload->getFileSize();
    $fileName= $upload->getFileName();
    $upload->setDestination(FILES_PATH . '/photo');
    $upload->receive();
}
```

Zend File

Thao tác upload với nhiều tập tin

```
if($this->request->isPost()){
    $upload = new \Zend\File\Transfer\Adapter\Http();

    // Mảng lưu trữ thông tin của một file upload
    $file01Info= $upload->getFileInfo('my_file_01');

    $upload->setDestination(FILES_PATH . '/photo', 'my_file_01');
    $upload->receive('my_file_01');
}
```

Kiểm tra điều kiện upload tập tin

```
if($this->request->isPost()){
    $upload = new \Zend\File\Transfer\Adapter\Http();
    $upload->addValidator('Extension', true, array('png', 'jpg'), 'picture');
    $upload->addValidator('Size', false, array('min' => '10Kb', 'max' => '8mb'), 'zip');
    $upload->setDestination(FILES_PATH . '/photo', 'my_file_01');
    $upload->setDestination(FILES_PATH . '/zip', 'my_file_02');

    $fileName = $upload->getFileName();
    foreach($fileName as $key => $value){
        if($upload->isValid($key)){
            $upload->receive($key);
        }else{
            $messages= $upload->getMessages();
            echo current($messages);
        }
    }
}
```

Sử dụng Filter

```
if($this->request->isPost()){
    $upload = new \Zend\File\Transfer\Adapter\Http();

    $upload->addValidator('Extension', true, array('doc', 'png', 'jpg'), 'my_file_01');
    $options= array(
        'target'=> FILES_PATH . '/photo/my_file_01.jpg',
        'overwrite' => true,
        'randomize' => true,
    );
    $upload->addFilter('Rename',$options,'my_file_01');

    if($upload->isValid('my_file_01')){
        $upload->receive('my_file_01');
    }else{
        $messages= $upload->getMessages();
        echo current($messages);
    }
}
```

HTML5 Form

Các loại input mới

- 1. Color
- 2. Date
- 3. Datetime
- 4. Datetime-local
- 5. Email
- 6. Month
- 7. Number
- 8. Range
- 9. Search
- 10. Tel
- 11. Time
- 12. Url
- 13. Week

HTML 5 Form

Các thuộc tính mới trong thẻ <form>

- autocomplete
- novalidate

HTML 5 Form

Các thuộc tính mới trong thẻ <input>

- autocomplete
- autofocus
- form
- formaction
- formenctype
- formmethod
- formnovalidate
- formtarget
- list
- min and max
- multiple
- pattern (regexp)
- placeholder
- required
- step

Zend Form

Zend\Form\Element

1. Button	9. DateTime	17. MonthSelect	25. Text
2. Captcha	10. DateTimeLocal	18. MultiCheckbox	26. Textarea
3. Checkbox	11. DateTimeSelect	19. Number	27. Time
4. Collection	12. Email	20. Password	28. Url
5. Color	13. File	21. Radio	29. Week
6. Csrf	14. Hidden	22. Range	
7. Date	15. Image	23. Select	
8. DateSelect	16. Month	24. Submit	

Zend Form

Zend\Form\View\Helper

1. Form	10. FormImage	19. FormSubmit	28. FormNumber
2. FormButton	11. FormInput	20. FormText	29. FormRange
3. FormCaptcha	12. FormLabel	21. FormTextarea	30. FormSearch
4. FormCheckbox	13. FormMultiCheckbox	22. FormColor	31. FormTel
5. FormCollection	14. FormPassword	23. FormDate	32. FormTime
6. FormElement	15. FormRadio	24. FormDateTime	33. FormUrl
7. FormElementErrors	16. FormReset	25. FormDateTimeLocal	34. FormWeek
8. FormFile	17. FormRow	26. FormEmail	
9. FormHidden	18. FormSelect	27. FormMonth	

Zend Form

Zend\Form\Element

1. Button	9. DateTime	17. MonthSelect	25. Text
2. Captcha	10. DateTimeLocal	18. MultiCheckbox	26. Textarea
3. Checkbox	11. DateTimeSelect	19. Number	27. Time
4. Collection	12. Email	20. Password	28. Url
5. Color	13. File	21. Radio	29. Week
6. Csrf	14. Hidden	22. Range	
7. Date	15. Image	23. Select	
8. DateSelect	16. Month	24. Submit	

Zend Form

Ràng buộc dữ liệu với InputFilter

- Sử dụng InputFilter để filter và validate dữ liệu trên form
- Review các class được hỗ trợ trong Zend Filter
- Review các class được hỗ trợ trong Zend Validator

Zend Form

Fieldset

- Tái sử dụng các element được sử dụng ở nhiều form khác nhau
- Định nghĩa filter và validator cho các element này thông qua `getInputFilterSpecification` của class `InputFilterProviderInterface`

Zend Form

Tùy biến giao diện thông báo lỗi (Dạng 1)

Login

Email

Dữ liệu không được rỗng!

Password

Dữ liệu không được rỗng!

Remember me

Sign in **Reset**

Not Registered? [Register here](#)

Zend Form

Tùy biến giao diện thông báo lỗi (Dạng 2)

Login

Email: Dữ liệu không được rỗng!
Password: Dữ liệu không được rỗng!

Email

Password

Remember me

Not Registered? [Register here](#)

Zend Form

Tùy biến giao diện thông báo lỗi (Dạng 3)

The image shows a login form with the following elements:

- Email:** A text input field with the placeholder "Email". It has a red border and a small red warning icon with an exclamation mark to its right.
- Password:** A text input field with the placeholder "Password".
- Remember me:** A checkbox labeled "Remember me".
- Buttons:** Two buttons at the bottom left: a green "Sign in" button and a grey "Reset" button.
- Error message:** A red tooltip-style box appears next to the Email field, containing the Vietnamese text "Đữ liệu không được rỗng!" (Data cannot be empty!).
- Footer:** At the bottom of the form, there is a link "Not Registered? Register here" in blue text.

Zend Form

Các vấn đề khác

- Create new element
- getFormElementConfig()
- bind()

Zend Form

Sử dụng Ajax trong Zend Framework (Load content)

php

zend

jquery

Zend Framework là một thư viện các lớp được xây dựng trên nền tảng ngôn ngữ PHP, theo hướng OOP và được công ty Zend phát triển. Zend Framework được định hướng theo mô hình MVC và là một PHP framework ra đời khá trễ, chính vì vậy ZF đã thừa hưởng những tinh hoa của các framework khác và tránh khỏi những sai lầm của các framework trước đã mắc phải trong quá trình phát triển.

Zend Form

Sử dụng Ajax trong Zend Framework (Selectbox)

Chọn lĩnh vực

Lập trình di động

Chọn khóa học

Chọn khóa học muốn đăng ký!

Submit

DATABASE

(tìm hiểu các thành phần Adapter, Sql, TableGateway, Pagination, ứng dụng Nested set model,)

Zend\Db\Adapter

Kết nối cơ sở dữ liệu

- Adapter là một thành phần quan trọng trong Zend\Db. Nó chịu trách nhiệm tạo mối liên hệ giữa các mã nguồn trong Zend\Db và hệ quản trị cơ sở dữ liệu

```
$adapter = new \Zend\Db\Adapter\Adapter(array
    'driver'      => 'Pdo_Mysql',
    'database'    => 'bookstore',
    'username'    => 'root',
    'password'    => '',
    'charset'     => 'utf8',
));
```

Zend\Db\Adapter

Thực hiện SQL đơn giản

- Cách 01

```
$result= $adapter->query('SELECT * FROM `user` WHERE `id` = ?', array(5));
```

- Cách 01

```
$sql = 'SELECT * FROM '
    . $adapter->platform->quoteIdentifier('user')
    . ' WHERE ' . $adapter->platform->quoteIdentifier('id') . ' = ' . $adapter-
    >driver->formatParameterName('id');

$statement = $adapter->query($sql);
$parameters = array('id' => 2);
$results = $statement->execute($parameters);
```

Zend\Db\Adapter

Đăng ký Adapter qua Servicemanager

- Khai báo mảng thông tin kết nối đến cơ sở dữ liệu
- Đăng ký qua ServiceManager

```
'service_manager'=> array(  
    'factories'=> array(  
        'Zend\Db\Adapter\Adapter' => 'Zend\Db\Adapter\AdapterServiceFactory',  
    ),  
)
```

- Sử dụng trong controller

```
$adapter = $this->getServiceLocator()->get('Zend\Db\Adapter\Adapter');
```

Zend\Db\Sql

Giới thiệu

- Zend\Db\Sql được dùng để xây dựng các câu lệnh truy vấn SQL. Kết quả trả về là tạo ra một Statement và Parameter container đại diện cho truy vấn đích, hoặc là một chuỗi SQL đầy đủ có thể thực thi được
 - \Zend\Db\Sql\Sql
 - \Zend\Db\Sql>Select
 - \Zend\Db\Sql\Insert
 - \Zend\Db\Sql\Update
 - \Zend\Db\Sql\Delete

Zend\Db\Sql

Các trường hợp truy vấn cần thực hiện

- Truy vấn từ một bảng dữ liệu nào đó
- Nhận kết quả trả về từ truy vấn
- Đặt tên mới cho bảng
- Xác định các cột cần hiển thị
- Đặt tên mới cho các cột
- Kết hợp các hàm xử lý chuỗi (tài liệu đính kèm)
- Kết hợp các hàm xử lý thời gian (tài liệu đính kèm)

Zend\Db\Sql

Các trường hợp truy vấn cần thực hiện

- Nhận kết quả truy vấn thông qua Zend\Db\Sql\Select
- Sử dụng order để sắp xếp danh sách
- Sử dụng limit và offset để tùy biến số phần tử, vị trí phần tử xuất hiện
- Sử dụng mệnh đề where đơn giản (WHERE ... AND ; WHERE ... OR ...)
- Sử dụng mệnh đề JOIN
- Sử dụng mệnh đề HAVING ... GROUP
- Loại bỏ thành phần nào đó trong câu lệnh truy vấn

Zend\Db\Sql

THỨ TỰ THỰC HIỆN CÁC MỆNH ĐỀ

SELECT ... FROM ...

WHERE ...

GROUP BY ... HAVING ...

ORDER BY ...

FROM → WHERE → GROUP BY → HAVING → SELECT → ORDER BY

Zend\Db\Sql

Kiểm tra dữ liệu với Zend\Db\Sql\Predicate trong mệnh đề WHERE

- \Zend\Db\Sql\Predicate\Between
- \Zend\Db\Sql\Predicate\Expression
- \Zend\Db\Sql\Predicate\In
- \Zend\Db\Sql\Predicate\IsNotNull
- \Zend\Db\Sql\Predicate\IsNull
- \Zend\Db\Sql\Predicate\Like
- \Zend\Db\Sql\Predicate\Literal
- \Zend\Db\Sql\Predicate\NotIn
- \Zend\Db\Sql\Predicate\NotLike
- \Zend\Db\Sql\Predicate\Operator
- \Zend\Db\Sql\Predicate\Predicate
- \Zend\Db\Sql\Predicate\PredicateSet

Zend\Db\Sql

Insert – Update – Delete

- \Zend\Db\Sql\Insert
- \Zend\Db\Sql\Update
- \Zend\Db\Sql\Delete

TableGateway

Xây dựng class database kế thừa từ AbstractTableGateway

- Xây dựng tập tin MenuTable.php thao tác với bảng dữ liệu menu
- Đăng ký qua ServiceManager
- Xây dựng các phương thức trong MenuTable.php để thực hiện các phương thức sau:
 - listItem()
 - getItem (id)
 - saveItem (insert , update)
 - deleteItem()

RecordExists & NoRecordExists

Kiểm tra sự tồn tại của dữ liệu

- \Zend\Validator\Db\RecordExists
- \Zend\Validator\Db\NoRecordExists
- Sử dụng tham số exclude

RecordExists & NoRecordExists

Bài tập tìm kiếm thông tin user theo địa chỉ email

 Find user

Email

ID	Username
1	john
3	jonathan

Zend\Paginator

Thông tin

- Thành phần rất linh hoạt giúp cho việc trình bày dữ liệu theo kiểu phân trang
- Các yếu tố cần lưu ý đến việc phân trang
 - Tổng số dòng (**record**) trong bảng (**totalItem**)
 - Số dòng hiển thị trên một trang (**ItemCountPerPage**)
 - Trang hiện tại (**currentPage**)
 - Số trang cần hiển thị (**pageRange**)

Manage user

List user

Add new user

#	Avatar	Username	Email	Fullscreen	Created	Action
4		CuGiai	cugiai@gmail.com	Cự Giải	16-10-2014	Info Edit Delete
2		KimNguu	kimnguu@gmail.com	Kim Ngưu	16-10-2014	Info Edit Delete
1		BachDuong	bachduong@gmail.com	Bạch Dương	16-10-2014	Info Edit Delete

« Previous 1 2 Next »

Manage user

Form add (edit)

Add user

Username	<input type="text"/>
Email	<input type="text"/>
Full name	<input type="text"/>

Submit **Cancel**

Manage user

Bổ sung chức năng

- Chức năng upload avatar
 - Cho phép upload avatar với định dạng png và jpg và resize kích thước hình ảnh phù hợp
 - Ở form edit nếu không thay đổi avatar vẫn giữ nguyên avatar cũ, trường hợp thay đổi avatar cần bảo đảm avatar cũ phải bị xóa
- Chức năng kích hoạt trạng thái tài khoản (status) của thành viên
 - Bổ sung selectbox cho phép chọn status của một user (form add và form edit)
 - Sử dụng ajax để thay đổi trạng thái tài khoản của một thành viên nào đó (index) có thông báo

ĐĂNG NHẬP – ĐĂNG KÝ

PHÂN QUYỀN

(tìm hiểu các thành phần Mail, Session, Captcha, Permission, ...)

Zend Captcha

Các vấn đề cần tìm hiểu

- Captcha là gì ?
- Sử dụng captcha Image trong Zend Framework
- Tìm hiểu các tham số về cấu hình captcha
- Validate captcha
- Xây dựng class Captcha và class Validate Captcha
- Kết hợp jQuery & Ajax xây dựng “Refresh Captcha”
- Captcha trong Zend Form

Zend Session

Các vấn đề cần tìm hiểu

- Khởi tạo session
- Lưu giá trị vào session (string, number, array, object, ...)
- Kiểm tra sự tồn tại của một session nào đó
- Lấy giá trị được lưu trong session
- Xóa session
- Thiết lập thời gian tồn tại của Session

Zend Mail

Các vấn đề cần tìm hiểu

- Một số định nghĩa
- Tìm hiểu phần cấu hình mail thông qua Zend\Mail\Transport\SmtpOptions
- Tìm hiểu phần thiết lập nội dung email qua lớp Zend\Mail\Message
- Tìm hiểu quản lý nội dung email với Zend\Mime\Mime
- Gửi mail với Zend\Mail\Transport\Smtp
- Sử dụng và tích hợp module về Mail (ZF2-AcMailer)

Zend Mail

Các định nghĩa

- Zend\Mail cung cấp chức năng soạn và gửi mail theo chuẩn MIME.
- MIME (Multipurpose Internet Mail Extensions) là một chuẩn Internet về định dạng cho thư điện tử. Hầu như mọi thư điện tử Internet được truyền qua giao thức SMTP theo định dạng MIME. (<http://vi.wikipedia.org/wiki/MIME>)
- Mail có thể được gửi thông qua các lớp: Transport\SendMail, Transport\Ssmtp, Transport\File

Zend Mail

Tích hợp & Sử dụng module ZF2-AcMailer



Alejandro Celaya
acelaya



Abdul Malik Ikhsan
samsonasik

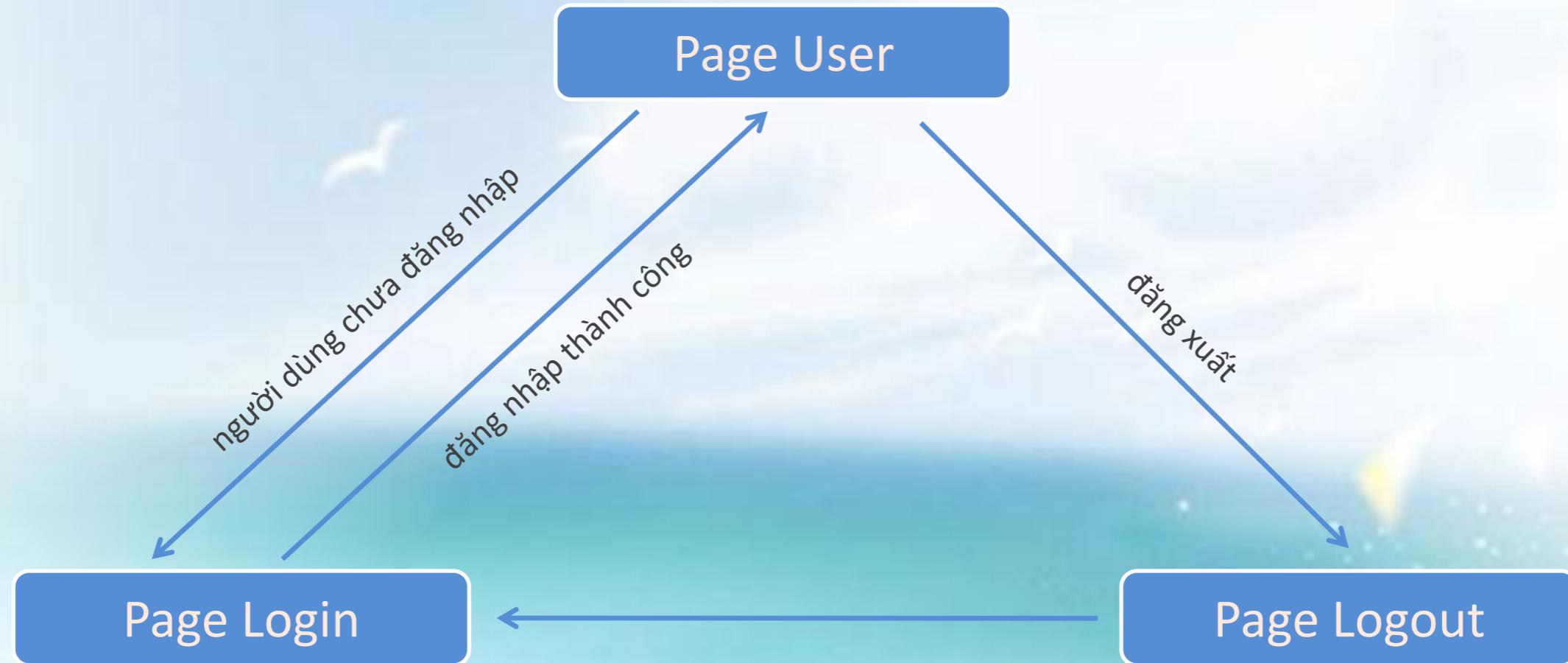
Zend Authenticate

Các vấn đề cần tìm hiểu

- Tìm hiểu cách sử dụng Authenticate
- Kiểm tra tài khoản đã được chứng thực hay chưa
- Lấy thông tin tài khoản
- Xóa chứng thực của tài khoản
- Sử dụng controller plugin Identity
- Đăng ký Authenticate qua SM
- Xây dựng class Authenticate

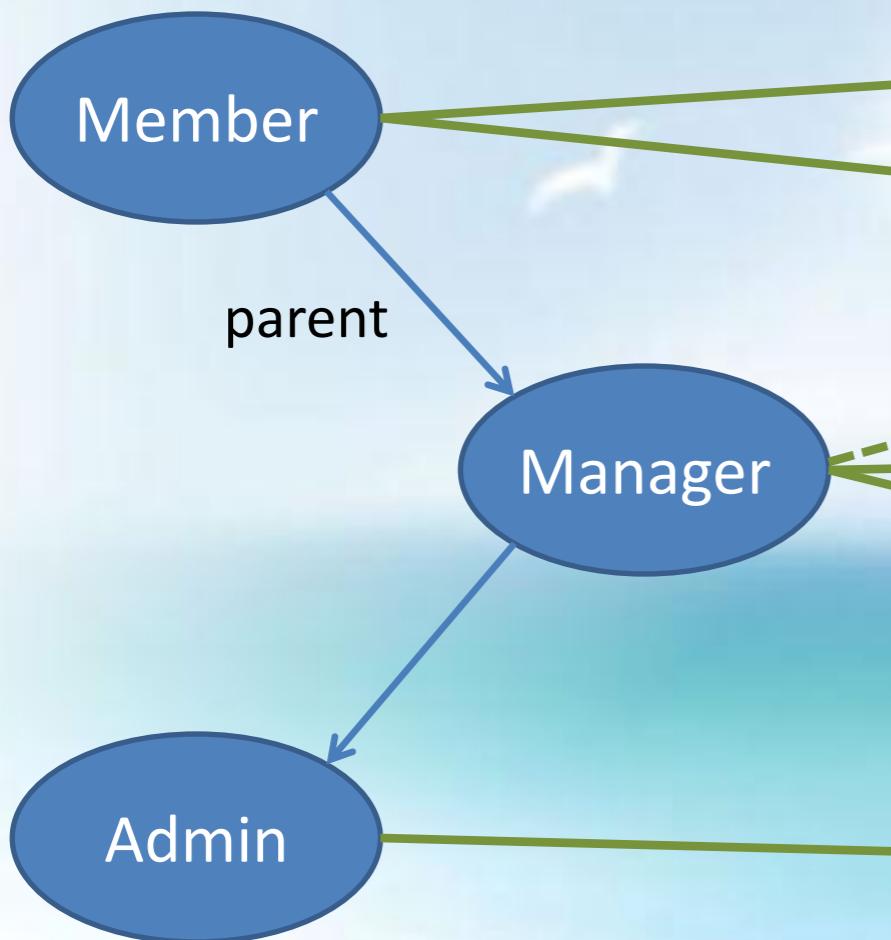
Zend Authenticate

Bài tập



Zend Permission

Zend\Permissions\Acl\Resource\GenericResource



allow

deny

allow

NewsController

index

info

add

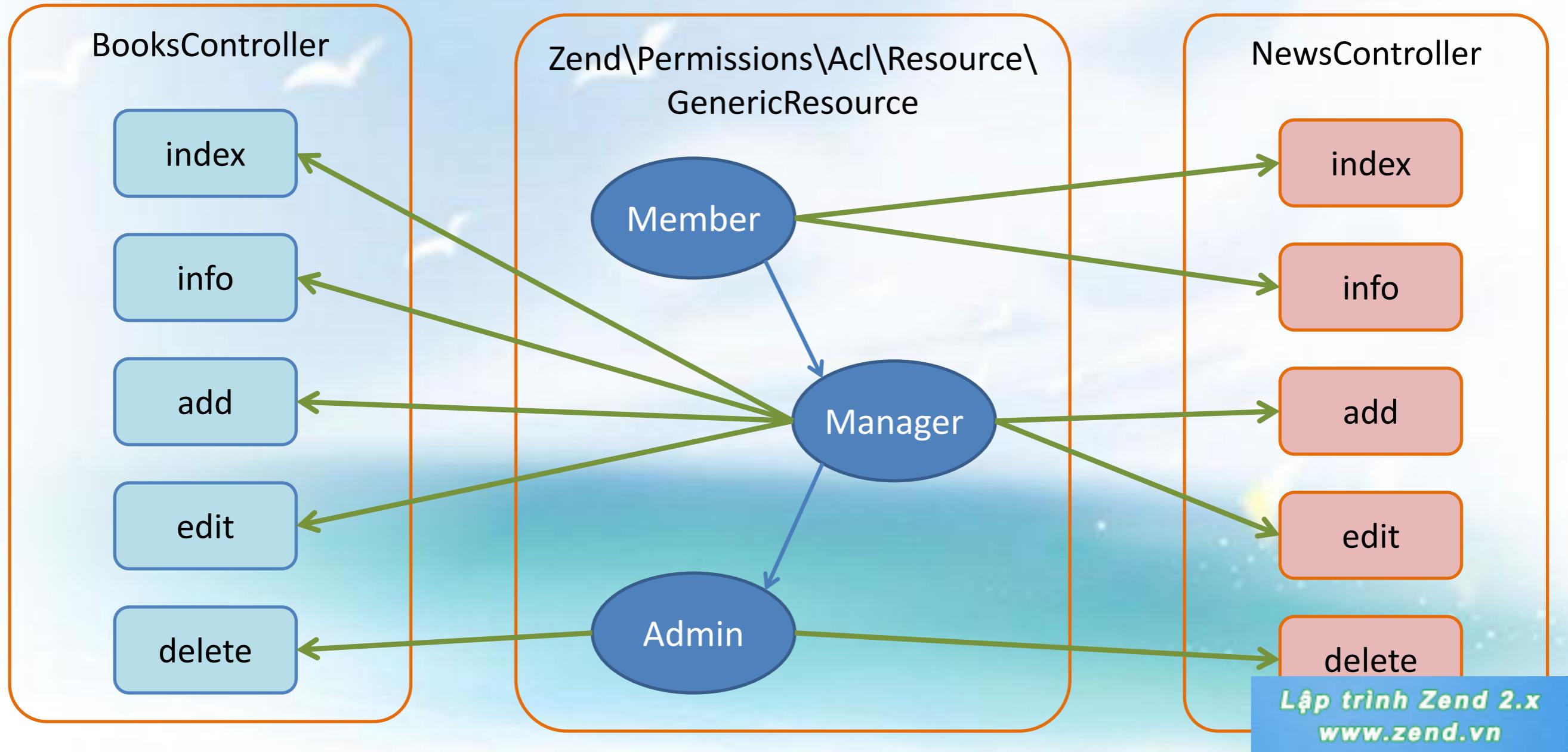
edit

delete

Phân quyền trên một controller trong một module

- Thiết lập danh sách các nhóm người dùng (Role) cần thực hiện thao tác phân quyền
- Xóa một người dùng nào đó ra khỏi danh sách (removeRole)
- Xóa tất cả dùng ra khỏi danh sách (removeRoleAll)
- Cấp phép truy cập
- Hủy cấp phép truy cập
- Từ chối truy cập
- Hủy từ chối truy cập
- Kiểm tra quyền truy cập ? (có hoặc không có quyền truy cập)

Phân quyền trên nhiều controller trong một module



Phân quyền trên nhiều controller trên nhiều module

Module DEF

UsersController

- index
- info
- add
- edit
- delete

GroupsController

- index
- info
- add
- edit
- delete

Zend\Permissions\Acl\Resource\GenericResource

Admin

Module ABC

NewsController

- index
- info
- add
- edit
- delete

BooksController

- index
- info
- add
- edit

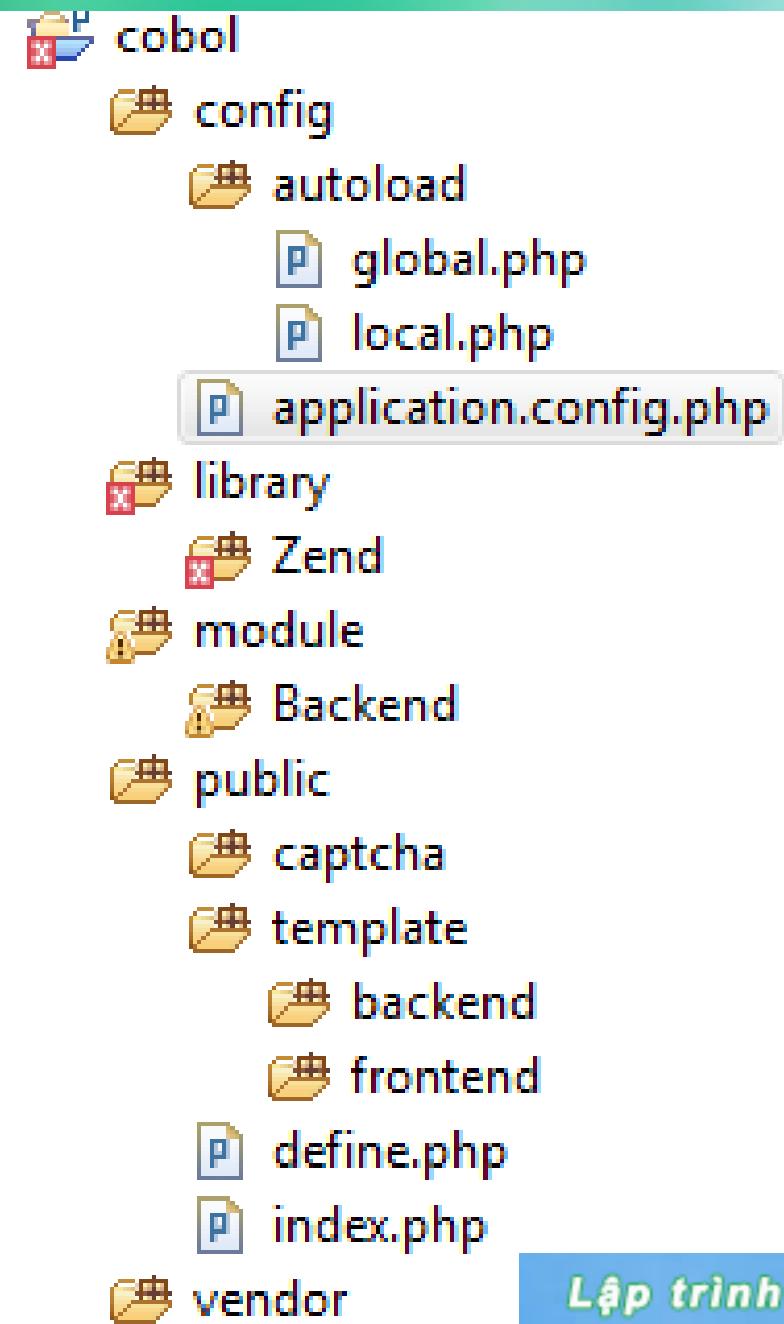
Áp dụng Zend Permission

- Module Permission: IndexController (index, info, deny)
- Module Database: IndexController (index, info)
- Role Admin có quyền truy cập Module Permission ở IndexController và indexAction
- Khi người dùng thuộc Role admin truy cập vào
 - Module: Permission – Controller: Index – Action: index → allow
 - Module: Permission – Controller: Index – Action: info → denyAction
 - Module: Database – Controller: Index – Action: index → denyAction
 - Module: Database – Controller: Index – Action: info → denyAction

Xây dựng cấu trúc ứng dụng

Xây dựng cấu trúc thư mục

- Xây dựng cấu trúc tập tin và thư mục
- Các tập tin application.config.php, global.php, local.php, để đơn giản trả về array rỗng
- Truy cập vào index.php để kiểm tra kết quả



Cấu hình và thiết lập virtualhost

xampp\apache\conf\extra\httpd-vhosts.conf

```
63 <VirtualHost *:80>
64     ServerAdmin postmaster@dummy-host2.localhost
65     DocumentRoot "D:/xampp/htdocs/cobol"
66     ServerName cobol.training.com
67     ##ServerAlias www.dummy-host2.localhost
68     ##ErrorLog "logs/dummy-host2.localhost-error.log"
69     ##CustomLog "logs/dummy-host2.localhost-access.log" combined
70 </VirtualHost>
```

C:\Windows\System32\drivers\etc

```
28 127.0.0.1 cobol.training.com
```

Định nghĩa các hằng số trong tập tin define.php

public/define.php

```
1 <?php
2
3     chdir(dirname(__DIR__));
4
5     define('APPLICATION_PATH', realpath(dirname(__DIR__)));
6     define('LIBRARY_PATH', APPLICATION_PATH . '/library');
7     define('HTMLPURIFIER_PREFIX', APPLICATION_PATH . '/vendor');
8     define('PUBLIC_PATH', APPLICATION_PATH . '/public');
9     define('FILES_PATH', PUBLIC_PATH . '/files');
10    define('CAPTCHA_PATH', PUBLIC_PATH . '/captcha');
11
12    define('APPLICATION_URL', 'http://training.zend2.com/');
13    define('PUBLIC_URL', APPLICATION_URL . '/public');
14    define('FILES_URL', PUBLIC_URL . '/files');
15    define('CAPTCHA_URL', PUBLIC_URL . '/captcha');
16
```

Xây dựng nội dung cho tập tin index.php

public/index.php

```
1 <?php
2     include_once 'define.php';
3     include_once LIBRARY_PATH . '/Zend/Loader/AutoloaderFactory.php';
4
5     Zend\Loader\AutoloaderFactory::factory(array(
6         'Zend\Loader\StandardAutoloader' => array(
7             'autoregister_zf' => true,
8         )
9     ));
10
11
12 if (!class_exists('Zend\Loader\AutoloaderFactory')) {
13     throw new RuntimeException('Zend\Loader\AutoloaderFactory is not exist!');
14 }
15
16 Zend\Mvc\Application::init(require 'config/application.config.php')->run();
```

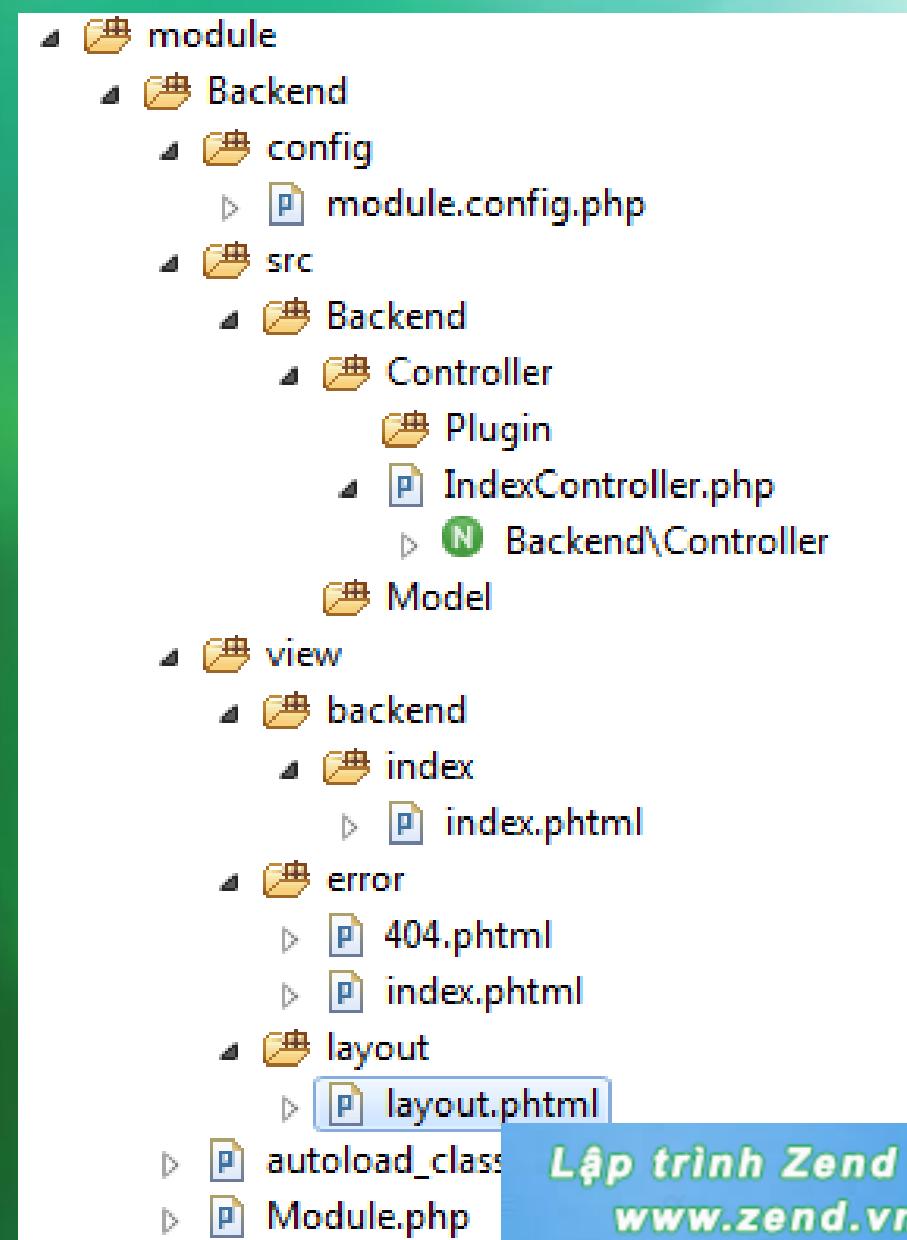
Cấu hình ứng dụng trong tập tin application.config.php

/config/application.config.php

```
1 <?php
2 return array(
3     'modules' => array(
4         'Backend',
5     ),
6     'module_listener_options' => array(
7         'module_paths' => array(
8             './module',
9             './vendor',
10        ),
11        'config_glob_paths' => array(
12            'config/autoload/{,*.}{global,local}.php',
13        ),
14    ),
15    'service_manager' => array(),
16 );
```

Xây dựng cấu trúc module Admin

- Xây dựng cấu trúc tập tin và thư mục như hình vẽ
- Xây dựng controller và tập tin view của action index trong controller đó
- Xây dựng tập tin Module.php chưa sử dụng autoload_classmap.php
- Xây dựng module.config.php



Xây dựng tập tin Module.php

/module/Backend/Module.php

```
<?php
namespace Backend;
use Zend\Mvc\ModuleRouteListener;
use Zend\Mvc MvcEvent;

class Module
{
    public function onBootstrap(MvcEvent $e)
    {
        echo '<h3 style="color:red;">' . __METHOD__ . '</h3>';
    }

    public function getConfig()
    {
        return include __DIR__ . '/config/module.config.php';
    }

    public function getAutoloaderConfig()
    {
        return array(
            'Zend\Loader\StandardAutoloader' => array(
                'namespaces' => array(
                    __NAMESPACE__ => __DIR__ . '/src/' . __NAMESPACE__,
                ),
            ),
        );
    }
}
```

Xây dựng tập tin module.config.php

/module/Backend/config/module.config.php

```
return array (
    'controllers' => array (
        'invokables' => array (
            'Backend\Controller\Index' => 'Backend\Controller\IndexController',
        )
    ),
    'view_manager' => array (
        'doctype' => 'HTML5',
        'display_not_found_reason' => true,
        'not_found_template' => 'error/404',
        'display_exceptions' => true,
        'exception_template' => 'error/index',
        'template_path_stack' => array(__DIR__ . '/../view'),
        'template_map' => array(
            'layout/layout' => __DIR__ . '/../view/layout/layout.phtml',
            'error/404' => __DIR__ . '/../view/error/404.phtml',
            'error/index' => __DIR__ . '/../view/error/index.phtml',
        ),
        'default_template_suffix' => 'phtml',
        'layout' => 'layout/layout'
    ),
    'router' => array (
        'routes' => array (
            'home' => array(
                'type' => 'Zend\Mvc\Router\Http\Literal',
                'options' => array (
                    'route' => '/',
                    'defaults' => array (
                        '__NAMESPACE__' => 'Backend\Controller',
                        'controller' => 'Backend\Controller\Index',
                        'action' => 'index'
                    )
                )
            )
        )
    )
);
```

Lưu ý khi router không hoạt động tốt !

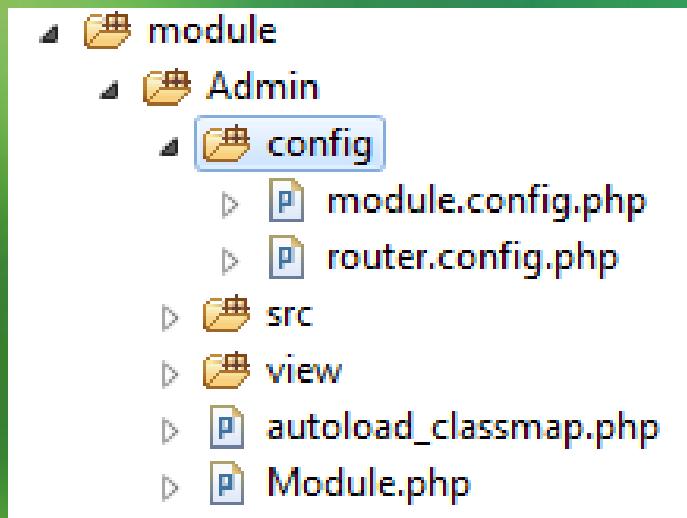
/module/Backend/Module.php

```
public function onBootstrap(MvcEvent $e)
{
    $eventManager          = $e->getApplication()->getEventManager();
    $moduleRouteListener = new ModuleRouteListener();
    $moduleRouteListener->attach($eventManager);

}
```

Tách phần cấu hình của router

Cấu trúc thư mục

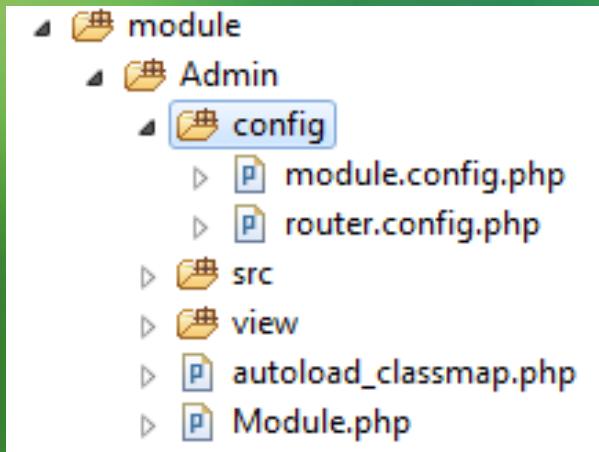


Cập nhật phương thức getConfig trong tập tin Module.php

```
public function getConfig()
{
    return array_merge(
        require_once __DIR__ . '/config/module.config.php',
        require_once __DIR__ . '/config/router.config.php'
    );
}
```

Xây dựng tập tin autoload_classmap.php

Cấu trúc thư mục

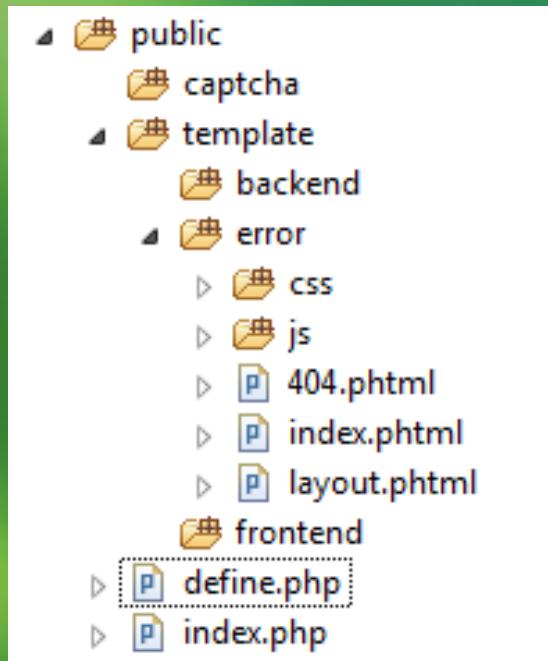


Cập nhật phương thức getAutoloaderConfig trong tập tin Module.php

```
public function getAutoloaderConfig()
{
    return array(
        'Zend\Loader\ClassMapAutoloader' => array(
            __DIR__ . '/autoload_classmap.php',
        ),
        'Zend\Loader\StandardAutoloader' => array(
            'namespaces' => array(
                __NAMESPACE__ => __DIR__ . '/src/' . __NAMESPACE__,
            ),
        ),
    );
}
```

Change layout error

Cấu trúc thư mục



Thay đổi layout đối với tập tin view

```
1 <?php $this->layout()->setTemplate('layout/error'); ?>
2 <h1><?php echo $this->translate('A 404 error occurred') ?></h1>
3 <h2><?php echo $this->message ?></h2>
4
5 <?php if (isset($this->reason) && $this->reason): ?>
6
```

Xóa giá trị public trong URL

1. Sử dụng .htaccess
2. Tạo thêm tập tin index.php “ảo”
3. Cấu hình lại virtualhost

(keyword: zend framework 2 remove public from url)

```
1 RewriteEngine On
2 RewriteCond %{REQUEST_FILENAME} -s [OR]
3 RewriteCond %{REQUEST_FILENAME} -l [OR]
4 RewriteCond %{REQUEST_FILENAME} -d
5 RewriteRule ^.*$ - [NC,L]
6
7 RewriteRule      ^.*$          public/index.php
8
```

Tích hợp thư viện ZendVN đã xây dựng

1. Đặt folder Zendvn cùng cấp với folder Zend
2. Điều chỉnh namespace cho phù hợp (phân biệt hoa – thường)
3. Kiểm tra một số class đã hoạt động tốt hay chưa

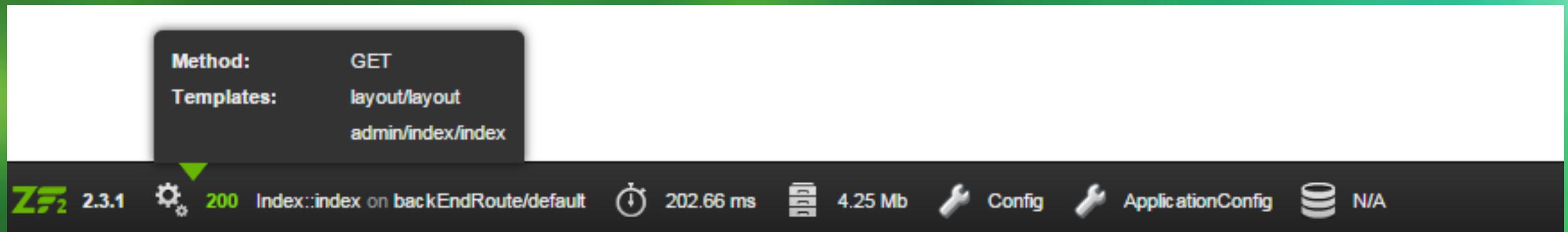
```
Zend\Loader\AutoloaderFactory::factory(array(
    'Zend\Loader\StandardAutoloader' => array(
        'autoregister_zf' => true,
        'namespaces'       => array(
            'ZendVN'        => LIBRARY_PATH . '/ZendVN'
        ),
    ),
));
});
```

Tích hợp module HTMLPurifier, Rtheadtitle, AcMailer



Sử dụng module DeveloperTool

Full document & source: <https://github.com/zendframework/ZendDeveloperTools>



Xây dựng giao diện Admin với Bootstrap 3



Xây dựng ứng dụng Bookstore (Backend)

Cấu trúc project & nhúng giao diện admin

- Xây dựng cấu trúc project
- Xây dựng hệ thống folder chứa các giao diện của ứng dụng
- Hướng dẫn thay đổi giao diện đối với mỗi module

Xây dựng sidebar, breadcrumb, header và dashboard

- Xây dựng header
- Xây dựng breadcrumb
- Xây dựng sidebar
- Xây dựng dashboard

Manage group

1. Cơ sở dữ liệu lưu trữ thông tin group
2. Kết nối cơ sở dữ liệu trong dự án
3. Hiển thị danh sách group
4. Phân trang
5. Sắp xếp danh sách
6. Lọc danh sách theo trạng thái của group
7. Lọc danh sách theo giá trị được tìm kiếm

Manage group

8. Thay đổi trạng thái (status) của một group
9. Thay đổi trạng thái (status) của nhiều group
10. Thay đổi thứ tự sắp xếp (ordering) của một hoặc nhiều group
11. Xóa một hoặc nhiều group
12. Xem thông tin chi tiết của một group
13. Thêm mới group (add)
14. Cập nhật thông tin của một group nào đó (edit)

Manage user

1. Cơ sở dữ liệu lưu trữ thông tin user
2. Xây dựng trang index với order, status (multi), paginator, delete (multi), search (keyword, status, group)
3. Xây dựng formAction với các lưu ý:
 - Thêm mới và chỉnh sửa thông tin của một group
 - Điều kiện username và email duy nhất
 - Chỉnh sửa thông tin nhưng không thay đổi mật khẩu

Sử dụng ImageWorkshop

- Download & document: <http://phpimageworkshop.com/>
- Tìm hiểu các thao tác trên hình ảnh: Cropping, Filter, Flip, Opacity, Resizing, Rotating, Writing
- Tạo watermark cho hình ảnh
- Lưu hình ảnh

User Avatar

- Cập nhật lại giao diện hiển thị danh sách người dùng
- Upload và lưu trữ hình ảnh gốc mà người dùng đã upload
- Resize hình ảnh (tỉ lệ ?)
- Xử lý trường hợp người dùng cập nhật thông tin user nhưng không thay đổi avatar
- Lưu ý trường hợp xóa tài khoản của một người dùng cần xóa luôn cả hình ảnh của người dùng.

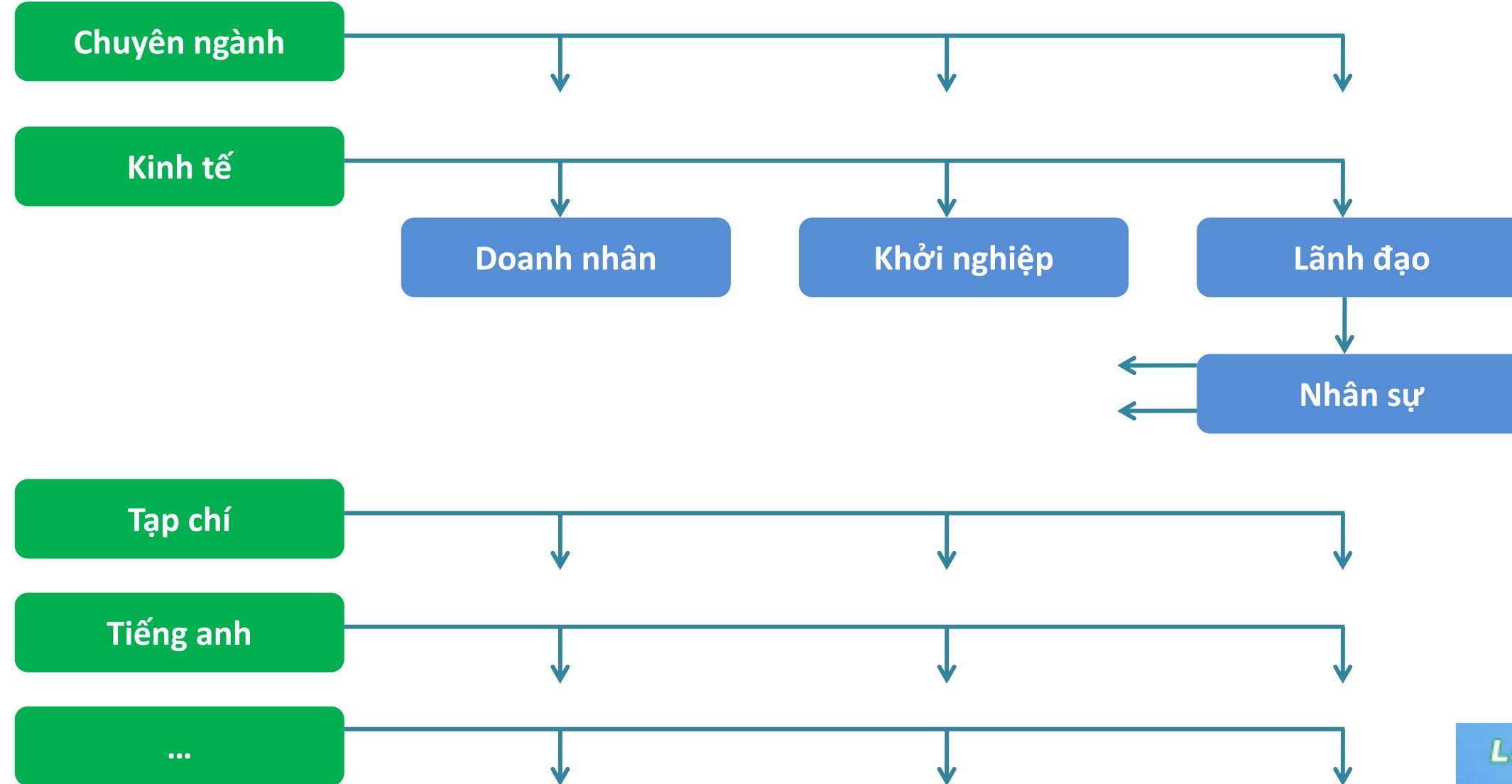
Sử dụng CKEditor và CKFinder

1. Download, cài đặt và nhúng CKEditor vào ứng dụng
2. Tùy chỉnh giao diện cho CKEditor: language, size, color, resize
3. Tùy chỉnh toolbar cho CKEditor: item, button, plugin, ...
4. Cấu hình CKEditor với ràng buộc dữ liệu
5. Cài đặt plugin và skin cho CKEditor
6. Tích hợp CKFinder vào CKEditor

Ứng dụng CKEditor với chữ ký của user

1. Tích hợp CKEditor vào project Zend 2
2. Kiểm tra dữ liệu hợp lệ
3. Lọc và hiển thị dữ liệu

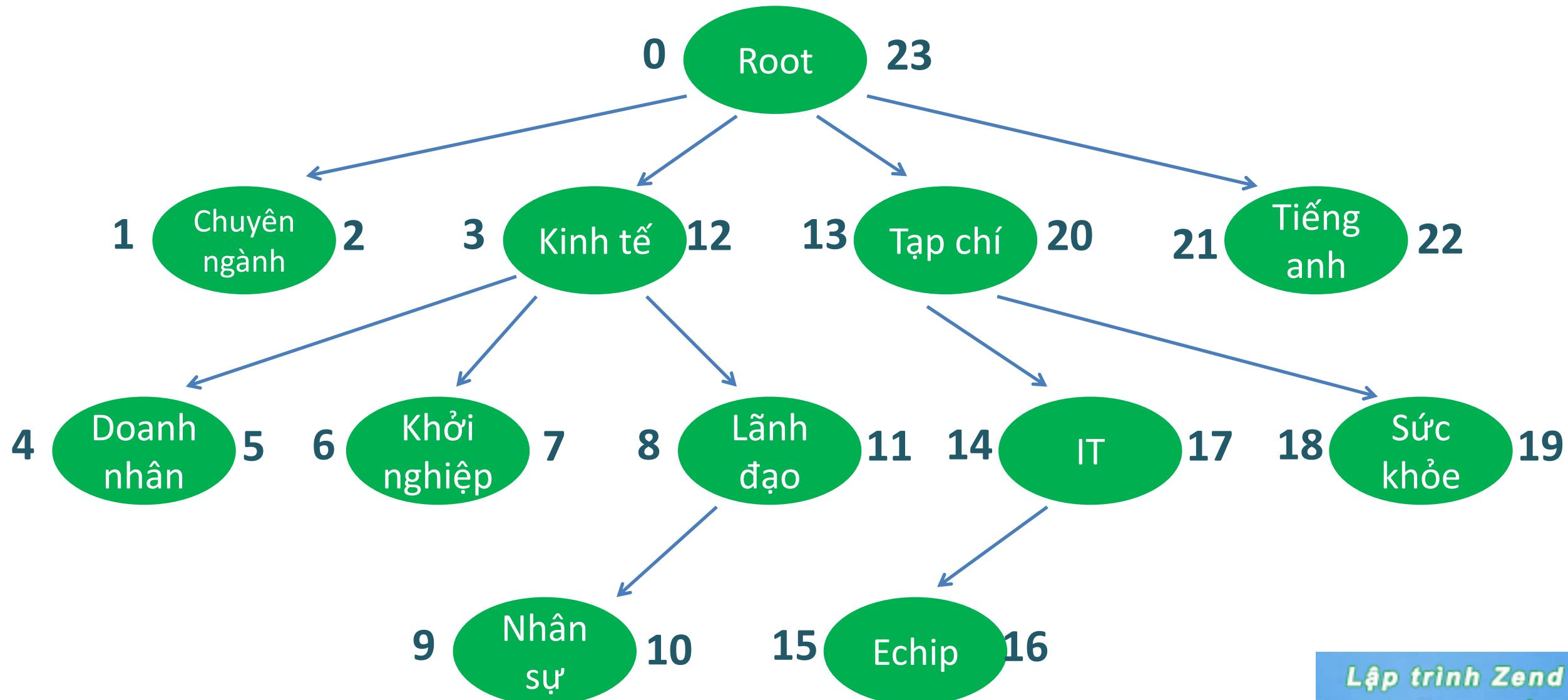
Quản trị hệ thống menu (category) đa cấp



Quản trị hệ thống menu (category) đa cấp

id	name	parent
1	Root	0
2	Chuyên ngành	1
3	Kinh tế	1
4	Tạp chí	1
5	Tiếng anh	1
6	Doanh nhân	3
7	Khởi nghiệp	3
9	Lãnh đạo	3
10	IT	4
11	Sức khỏe	4
12	Nhân sự	9
13	Echip	10

Quản trị hệ thống menu (category) đa cấp



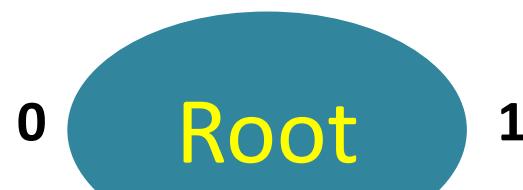
Quản trị hệ thống menu (category) đa cấp

id	name	status	parent	level	left	right
1	Root	1	0	0	0	23
2	Chuyên ngành	1	1	1	1	2
3	Kinh tế	1	1	1	3	12
4	Tạp chí	1	1	1	13	20
5	Tiếng anh	1	1	1	21	22
6	Doanh nhân	1	3	2	4	5
7	Khởi nghiệp	1	6	2	6	7
8	Lãnh đạo	1	1	2	8	11
9	IT	1	1	2	14	17
10	Sức khỏe	1	1	2	18	19
11	Nhân sự	1	1	3	9	10
12	Echip	1	1	3	15	16

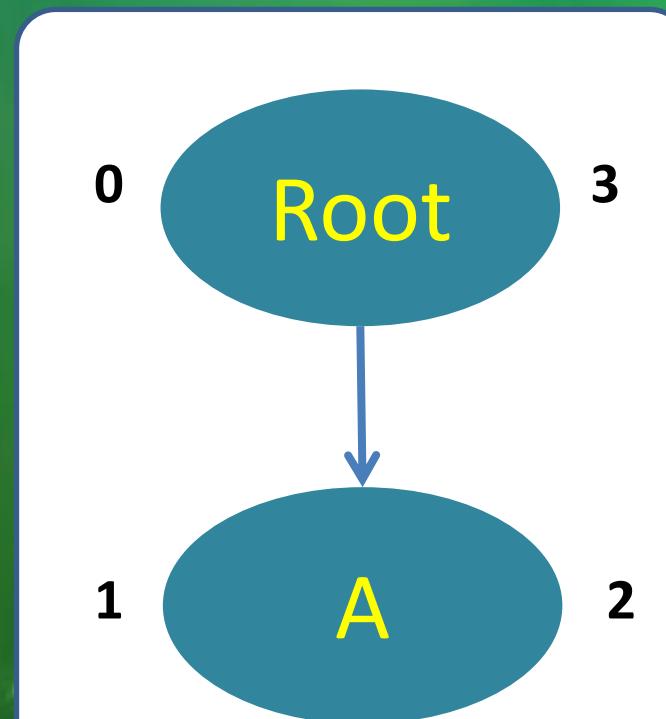
Nested Set Model

- Hiển thị toàn bộ cây menu (category) có phân cấp
- Hiển thị cây menu (category) theo level
- Hiển thị nhánh cây menu (category)
- Hiển thị đường đi đến một nút nào đó (breadcrumb)

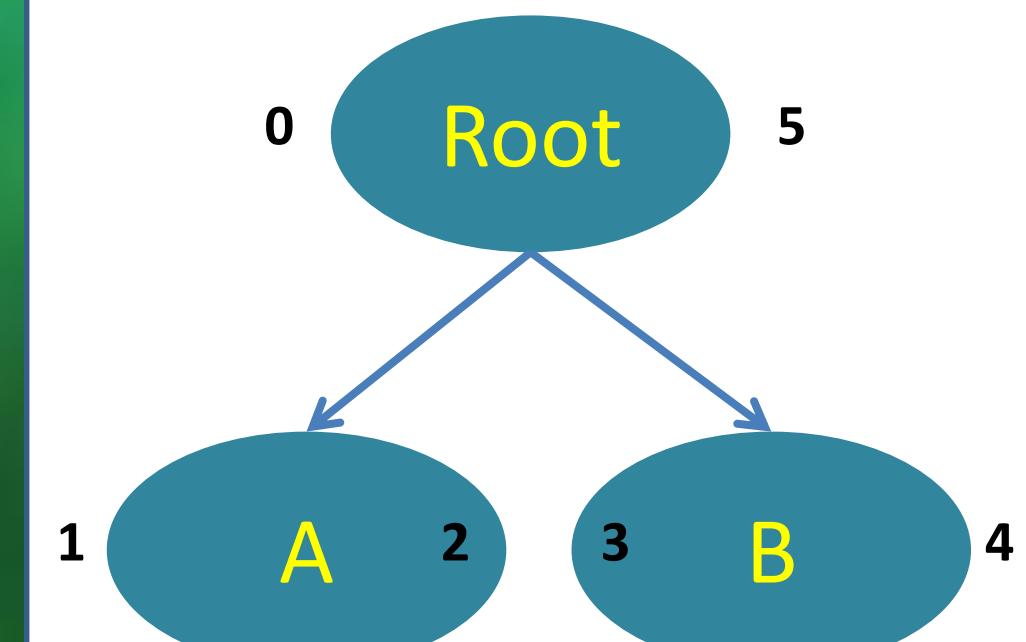
Nested Set Model



Nút gốc

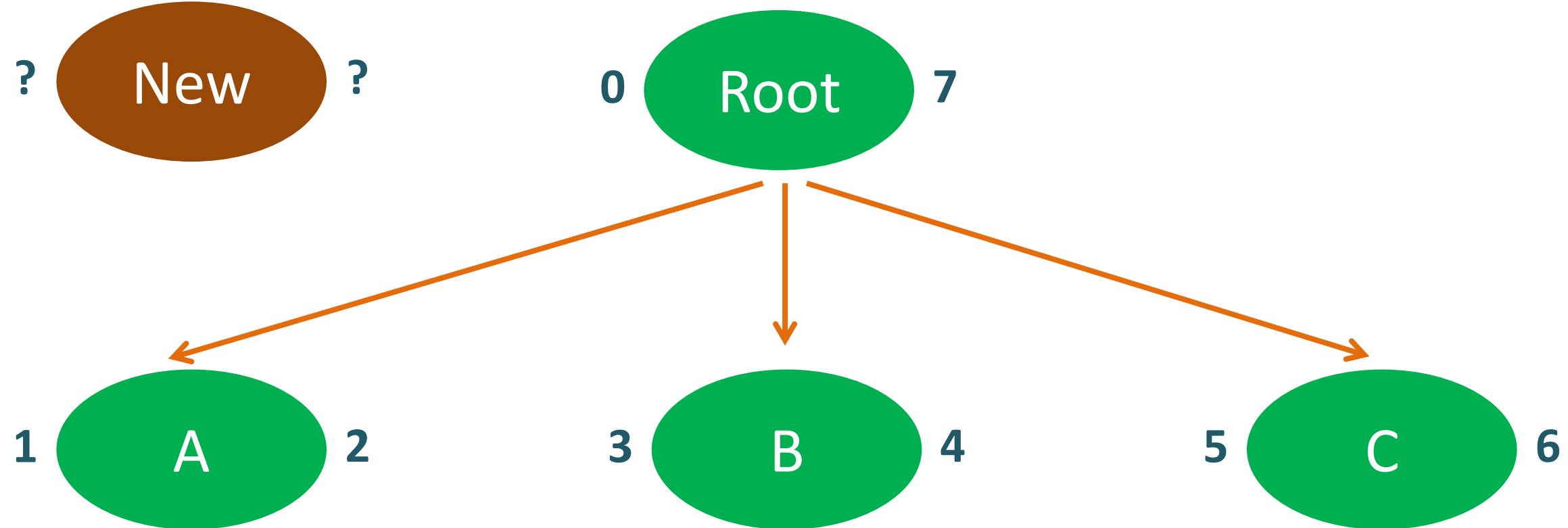


Nút gốc có 1 con

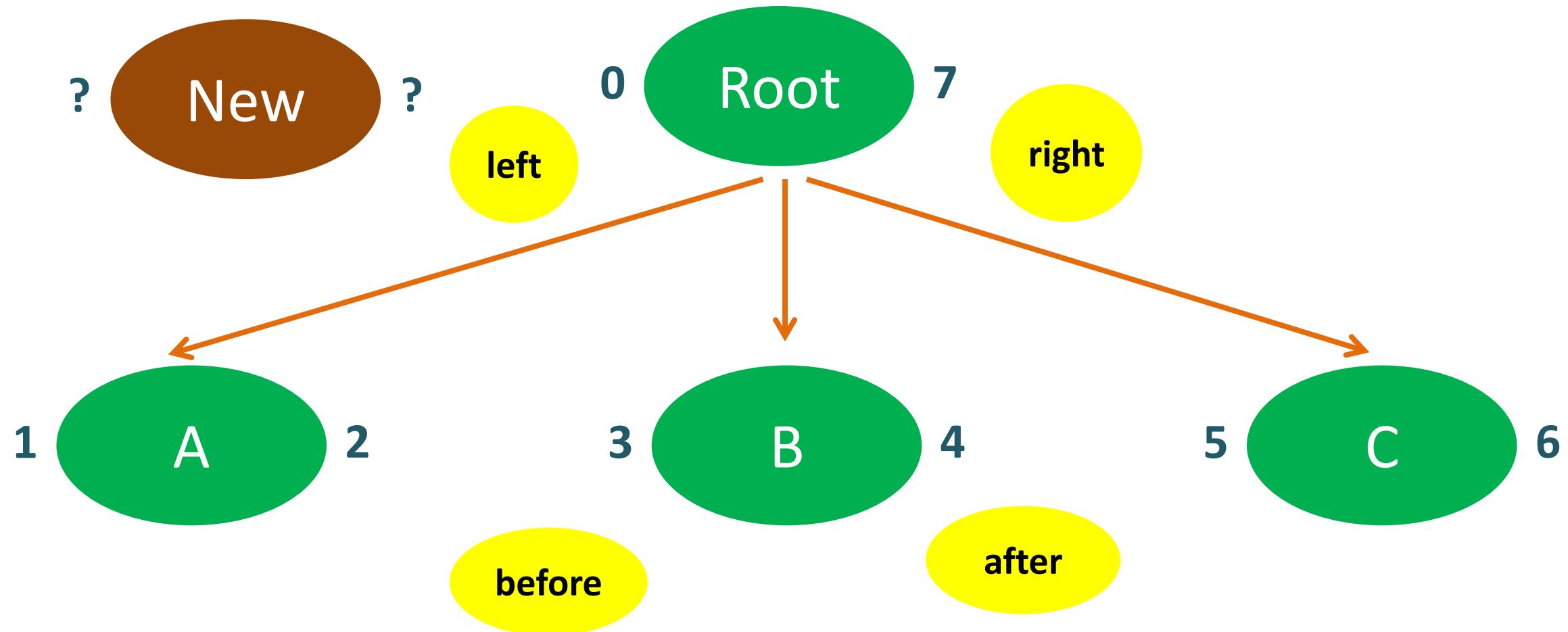


Nút gốc có 2 con

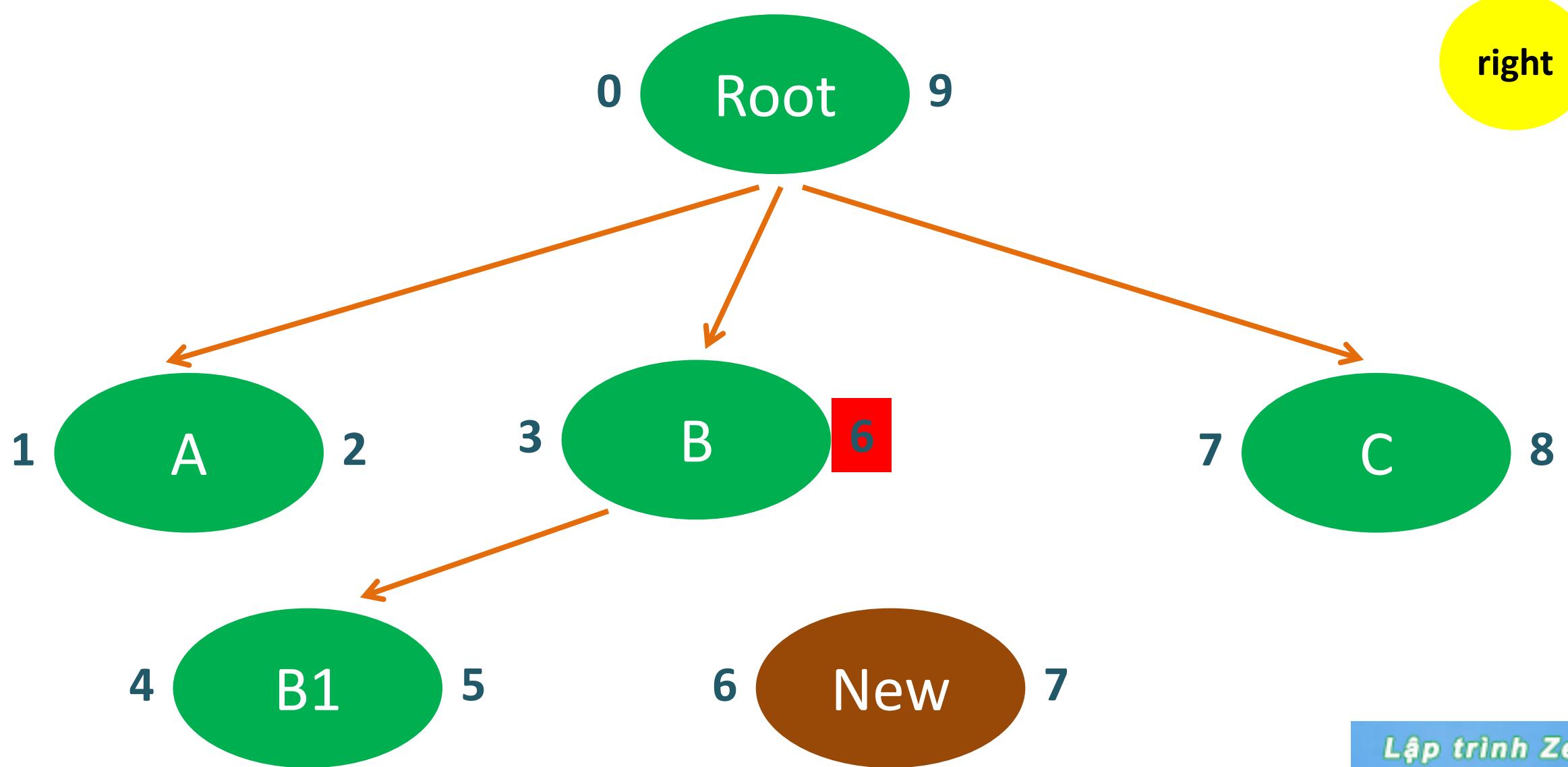
4 trường hợp thêm node vào cây



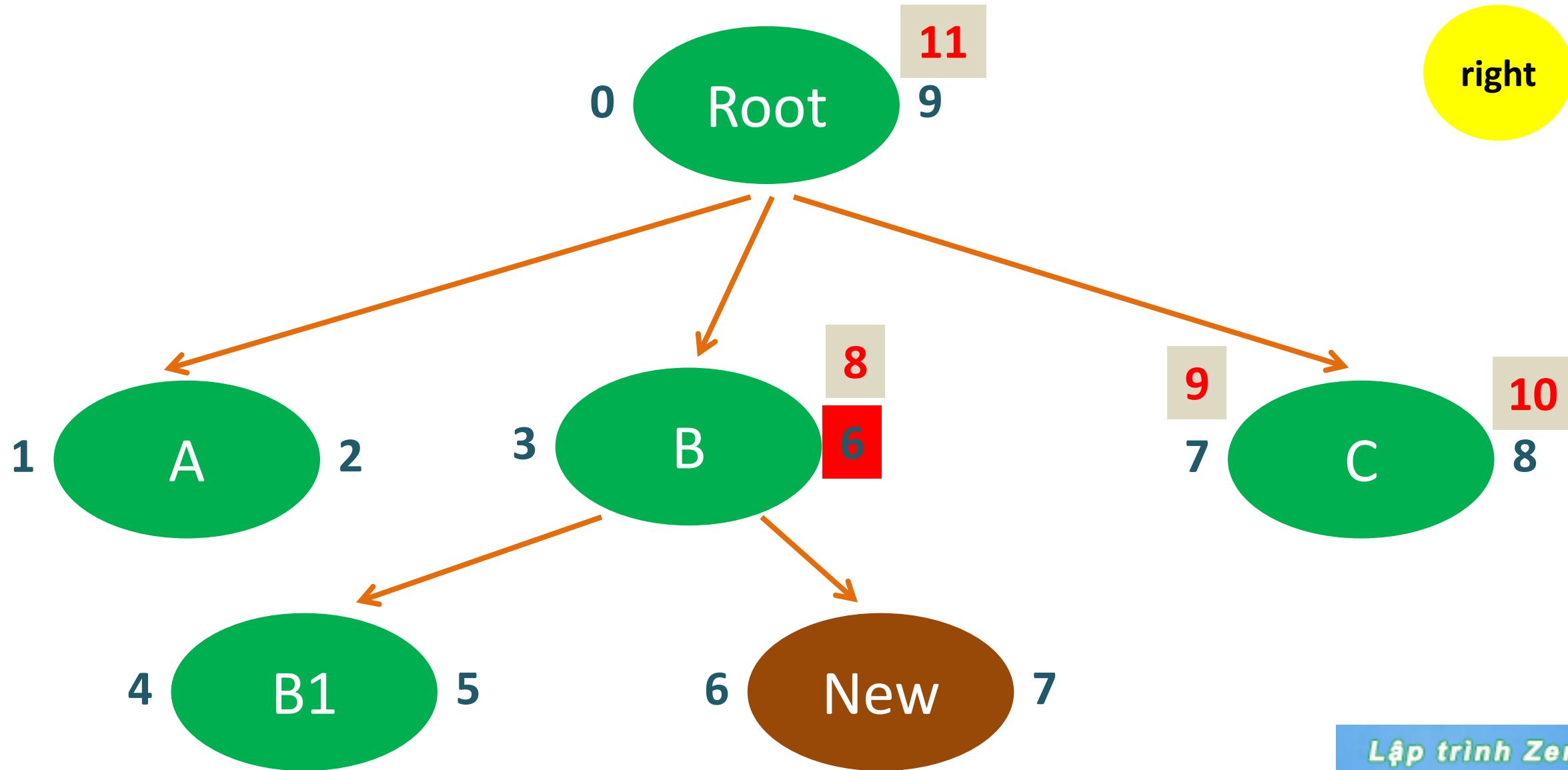
4 trường hợp thêm node vào cây



4 trường hợp thêm node vào cây (right)

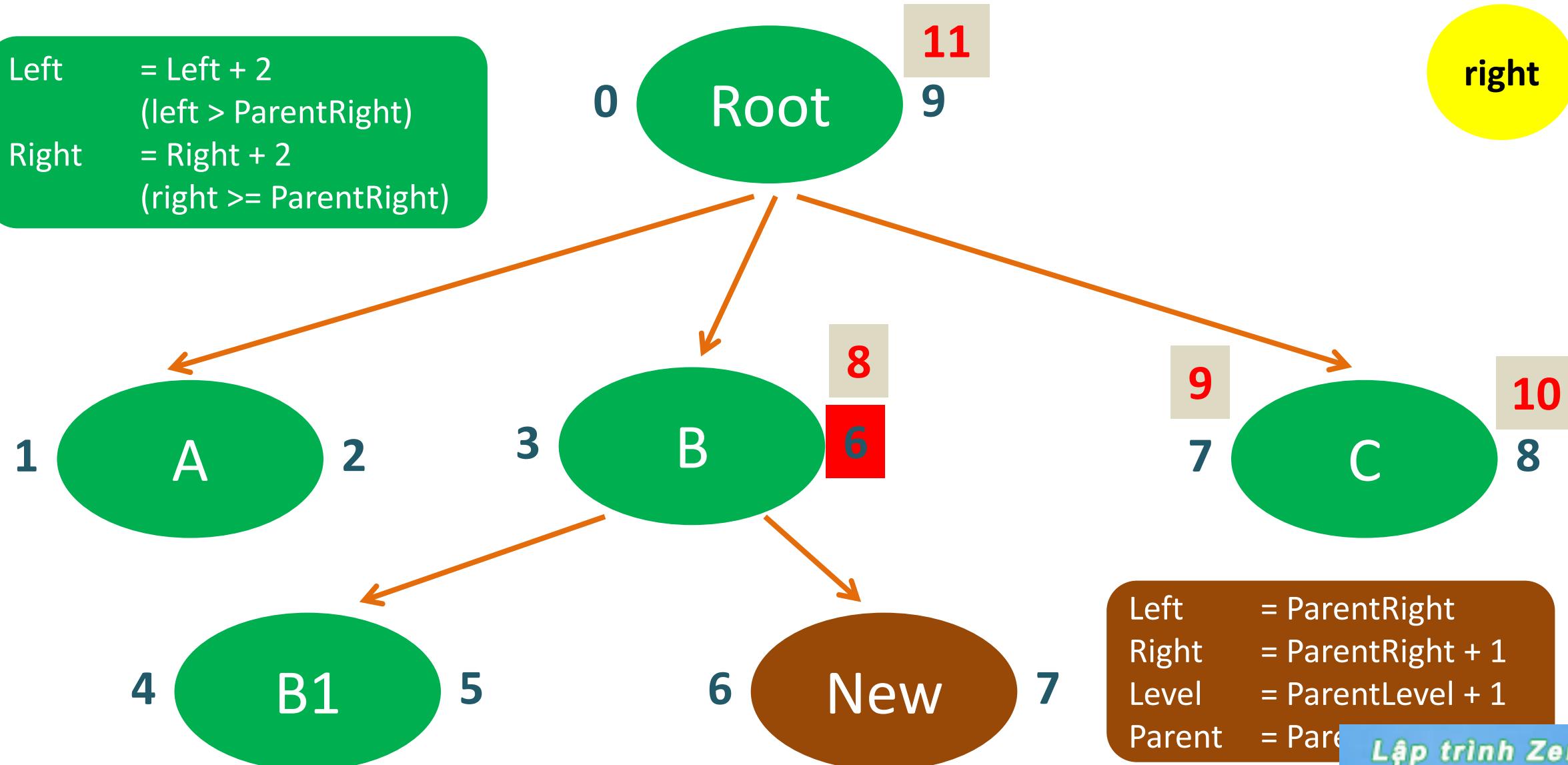


4 trường hợp thêm node vào cây (right)

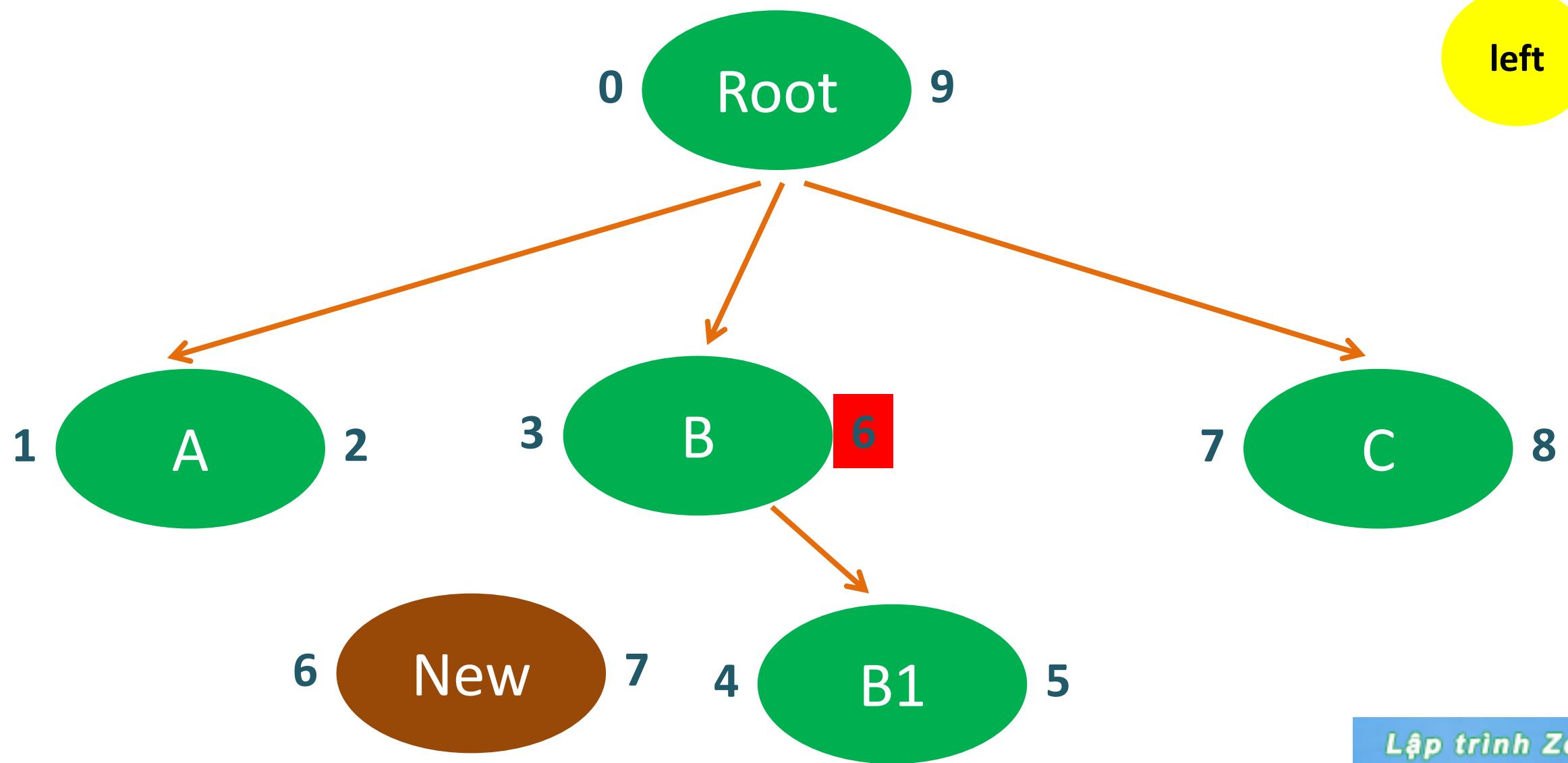


4 trường hợp thêm node vào cây (right)

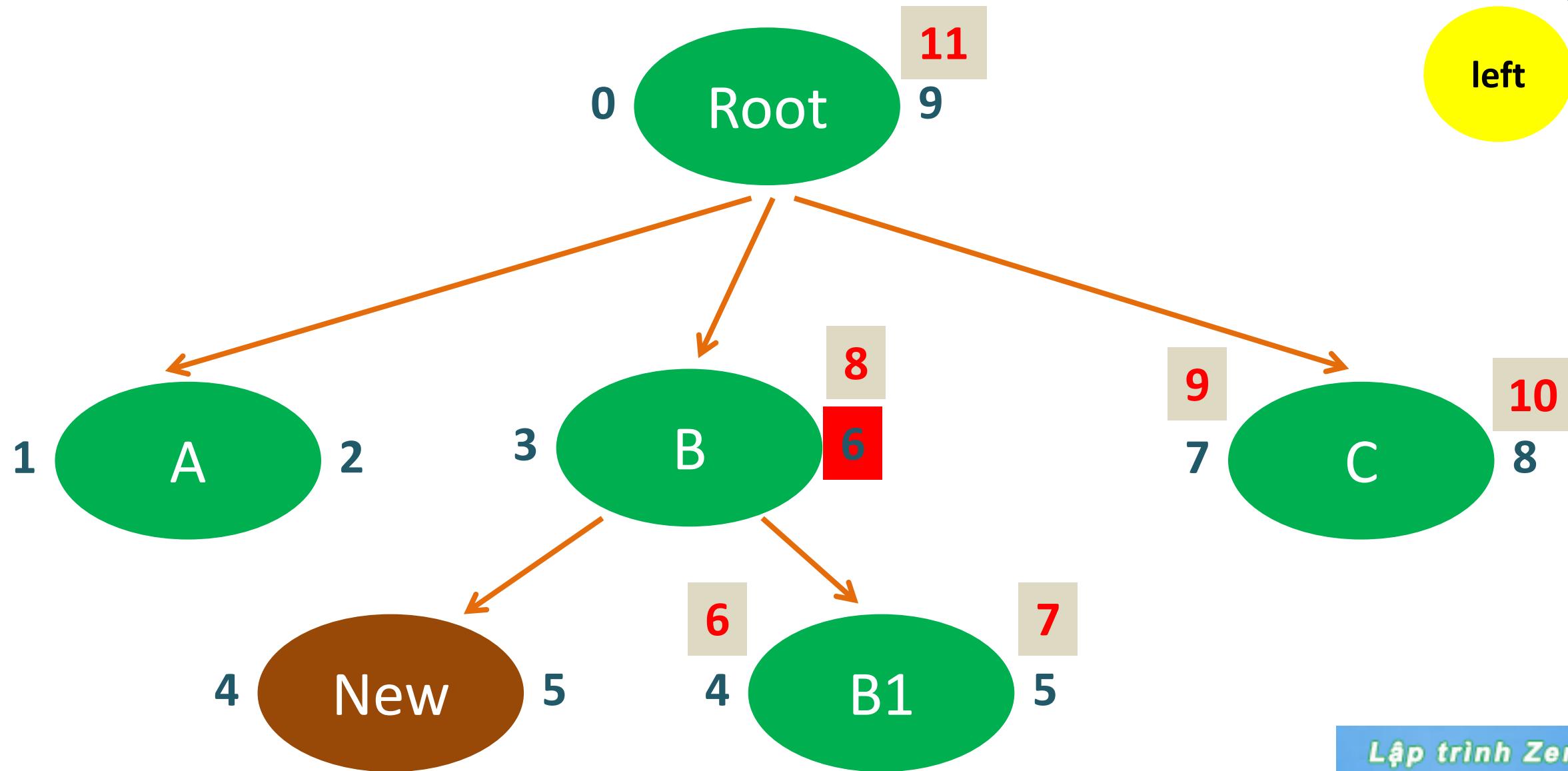
Left = Left + 2
(left > ParentRight)
Right = Right + 2
(right >= ParentRight)



4 trường hợp thêm node vào cây (left)

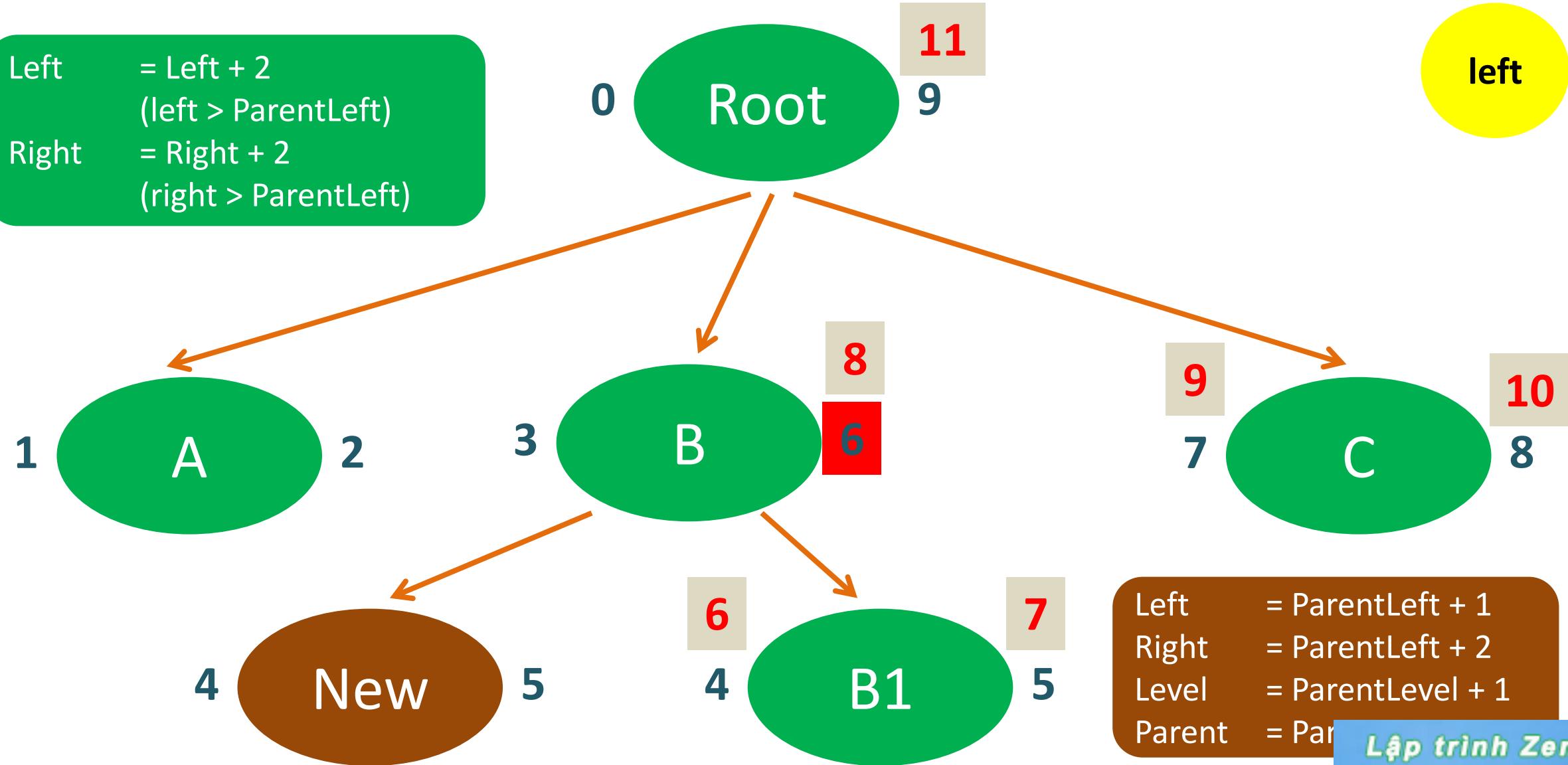


4 trường hợp thêm node vào cây (left)

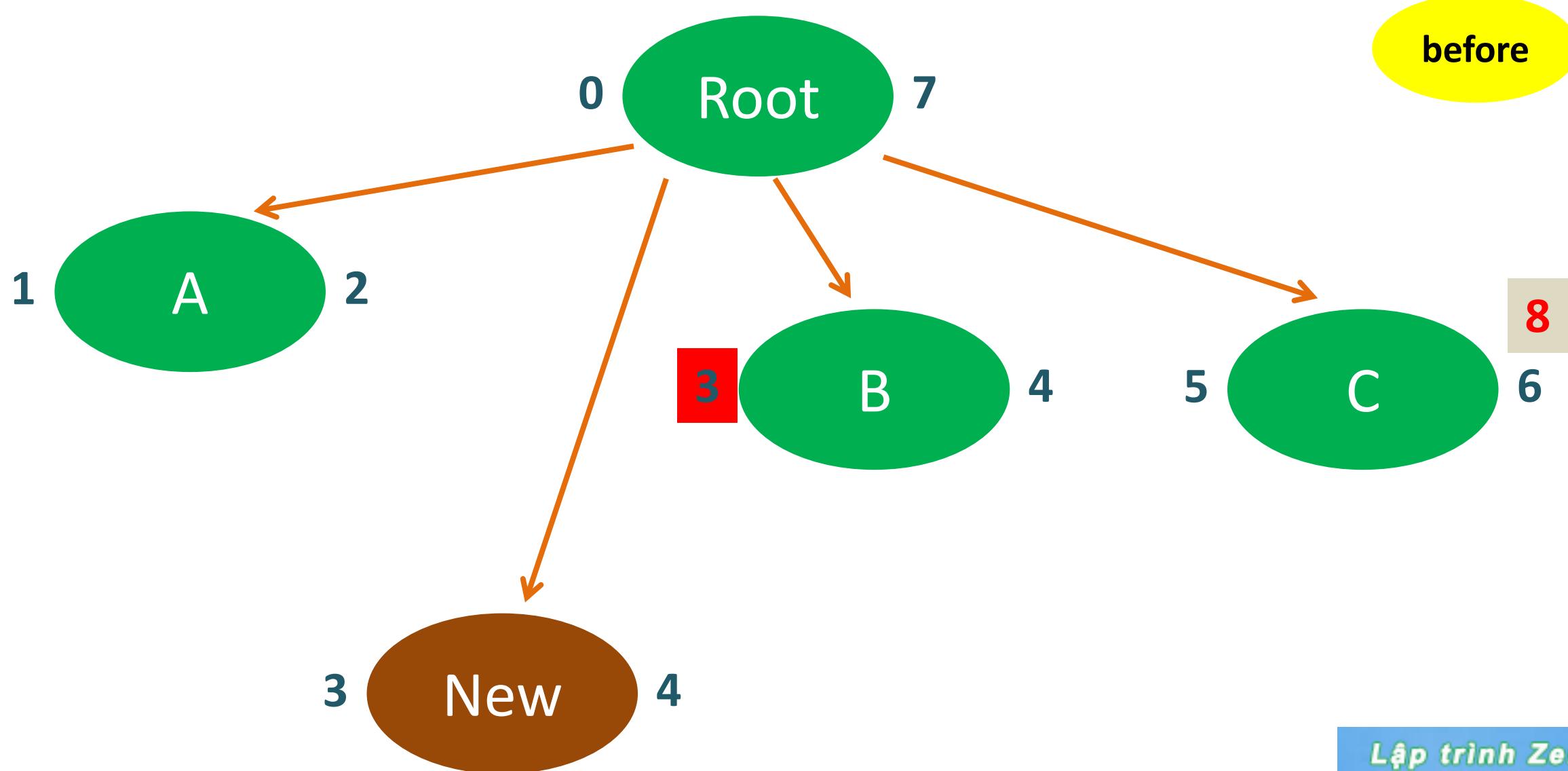


4 trường hợp thêm node vào cây (left)

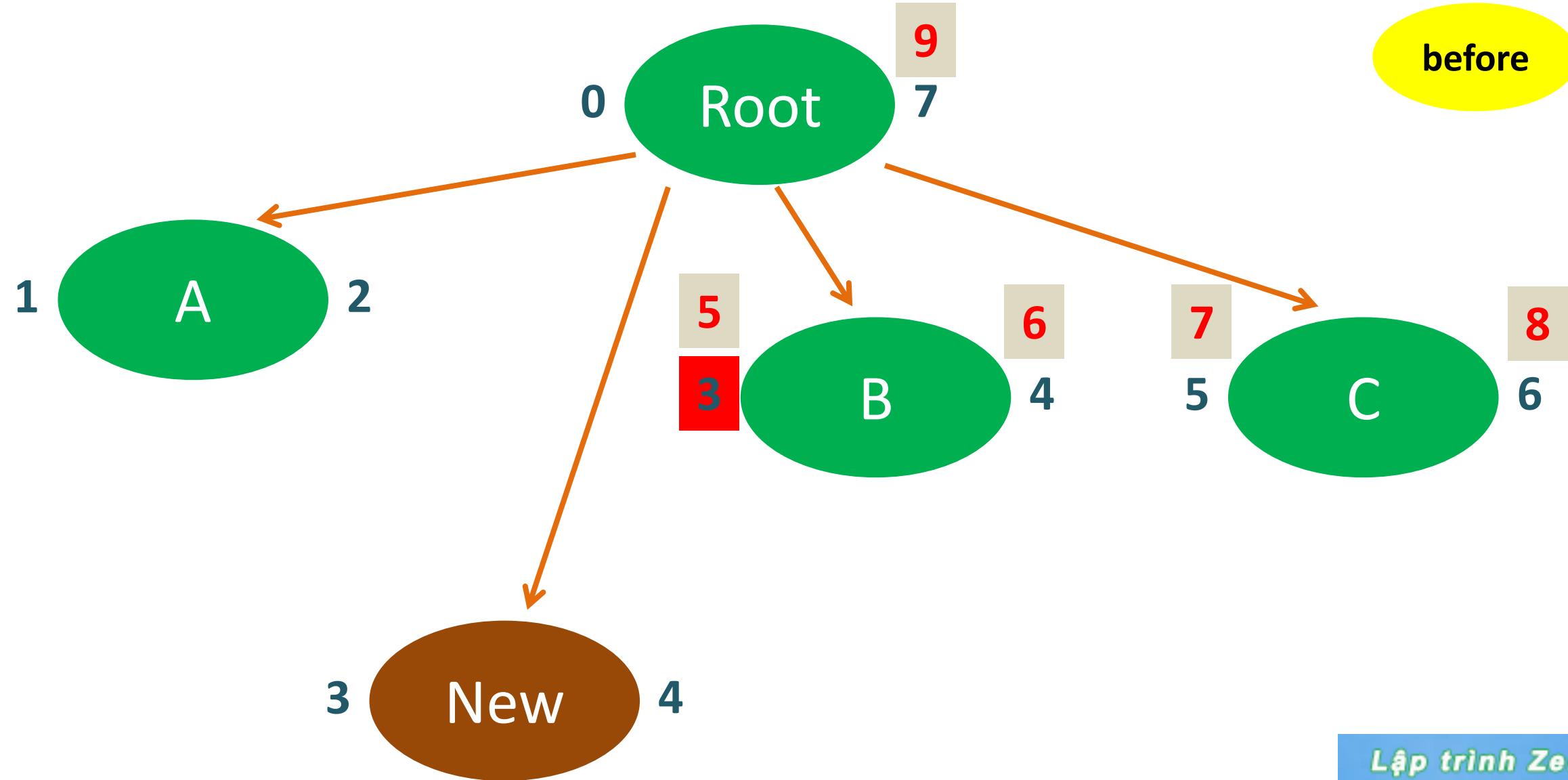
Left = Left + 2
(left > ParentLeft)
Right = Right + 2
(right > ParentLeft)



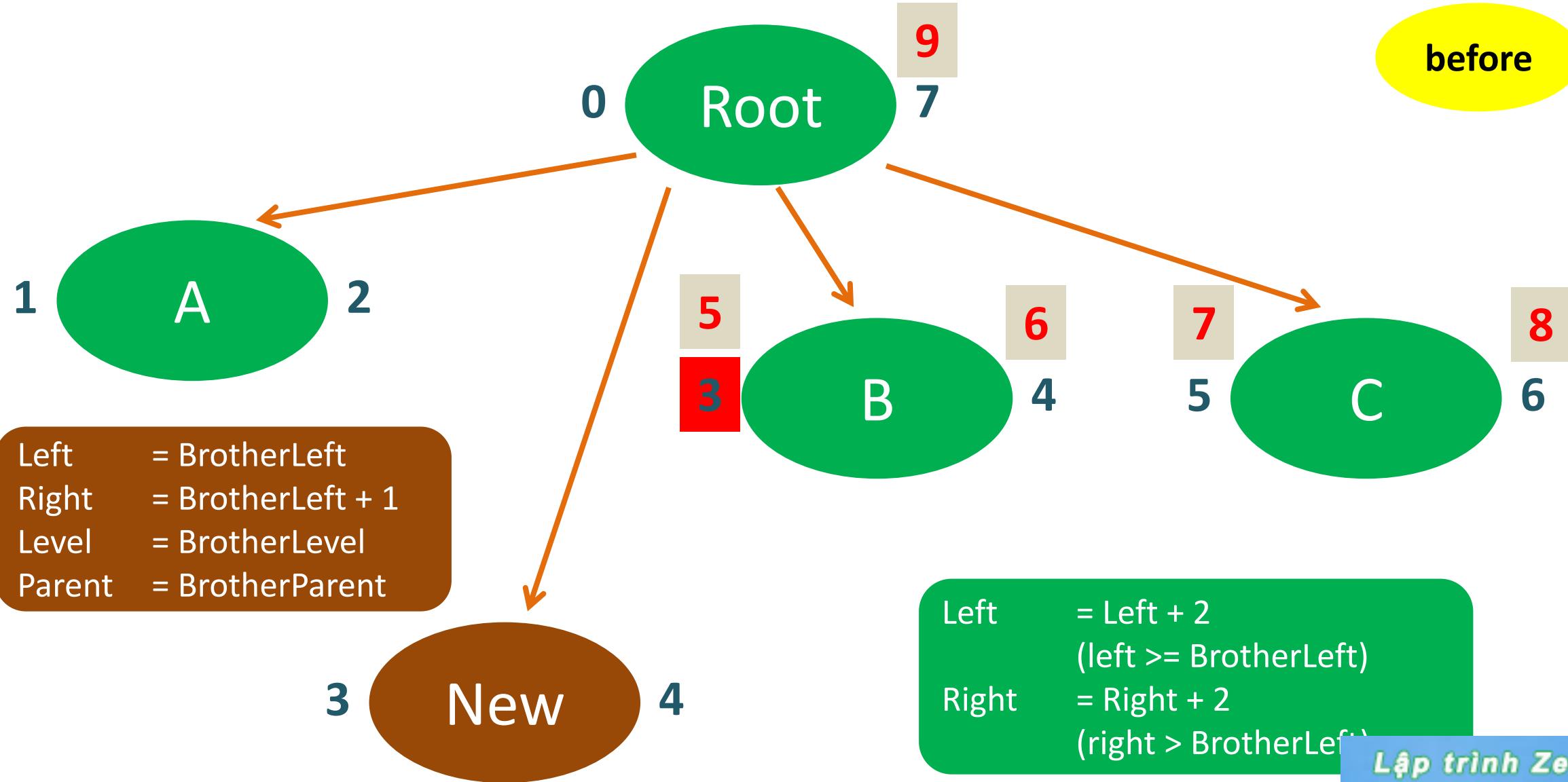
4 trường hợp thêm node vào cây (before)



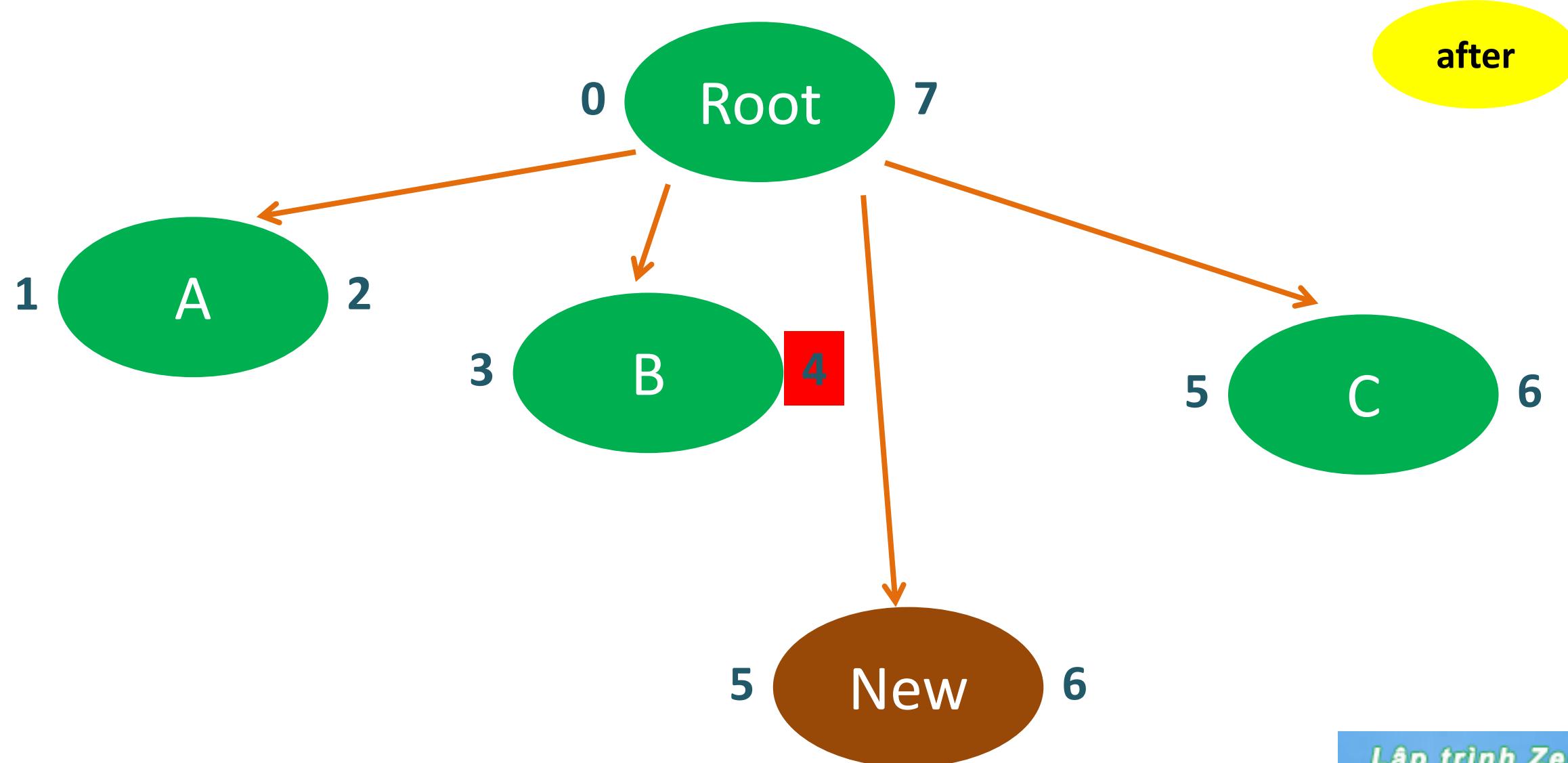
4 trường hợp thêm node vào cây (before)



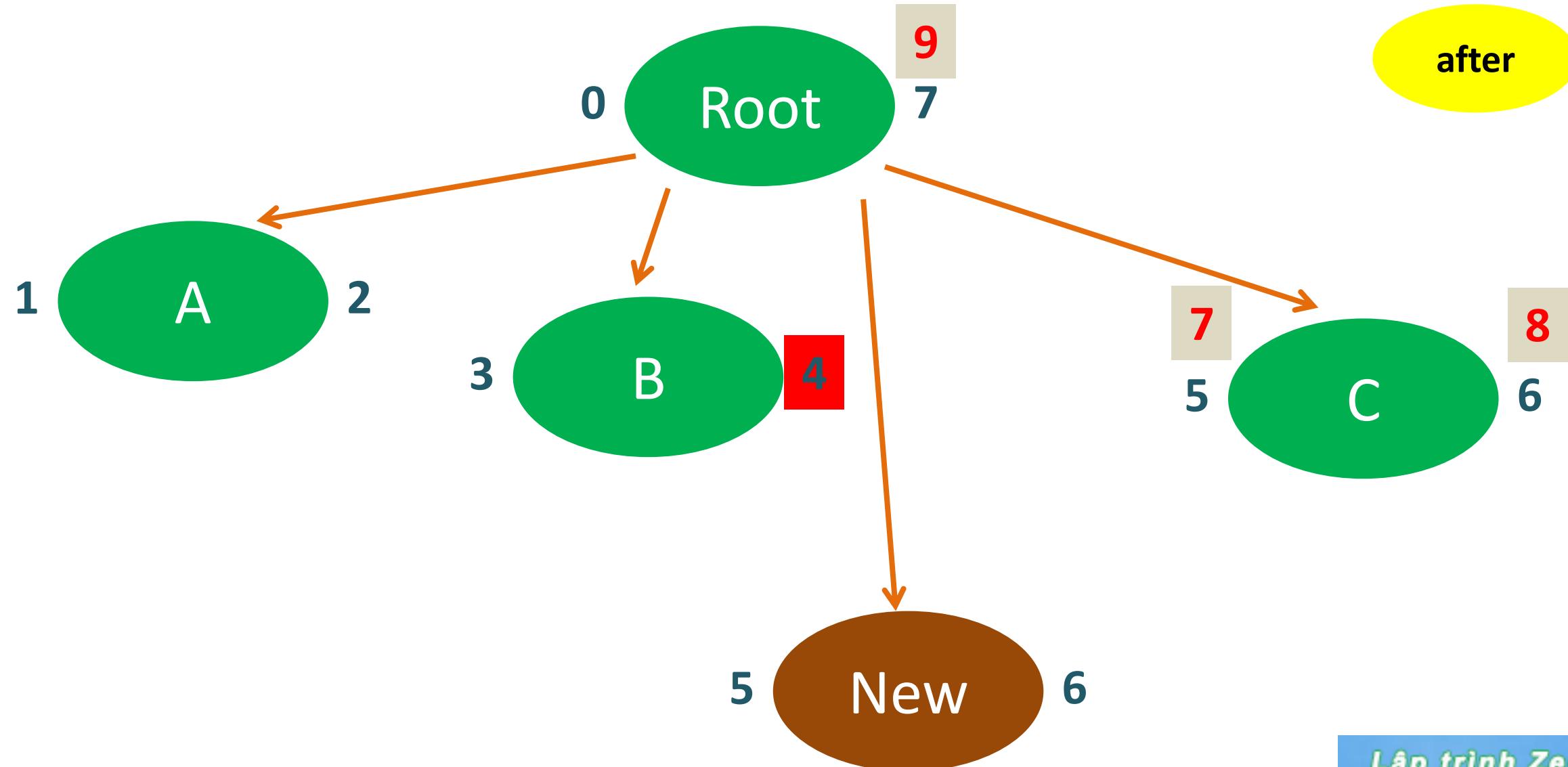
4 trường hợp thêm node vào cây (before)



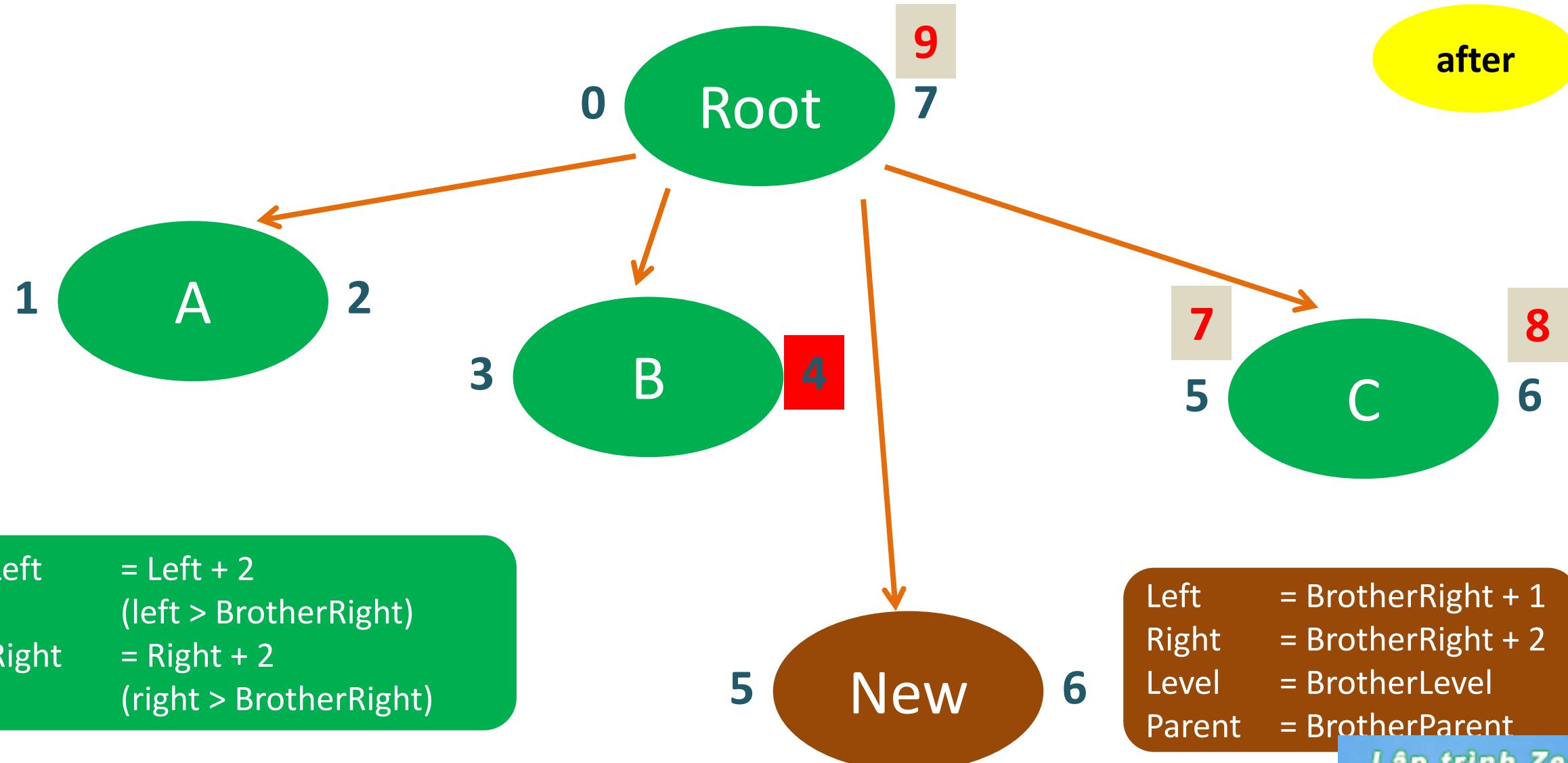
4 trường hợp thêm node vào cây (after)



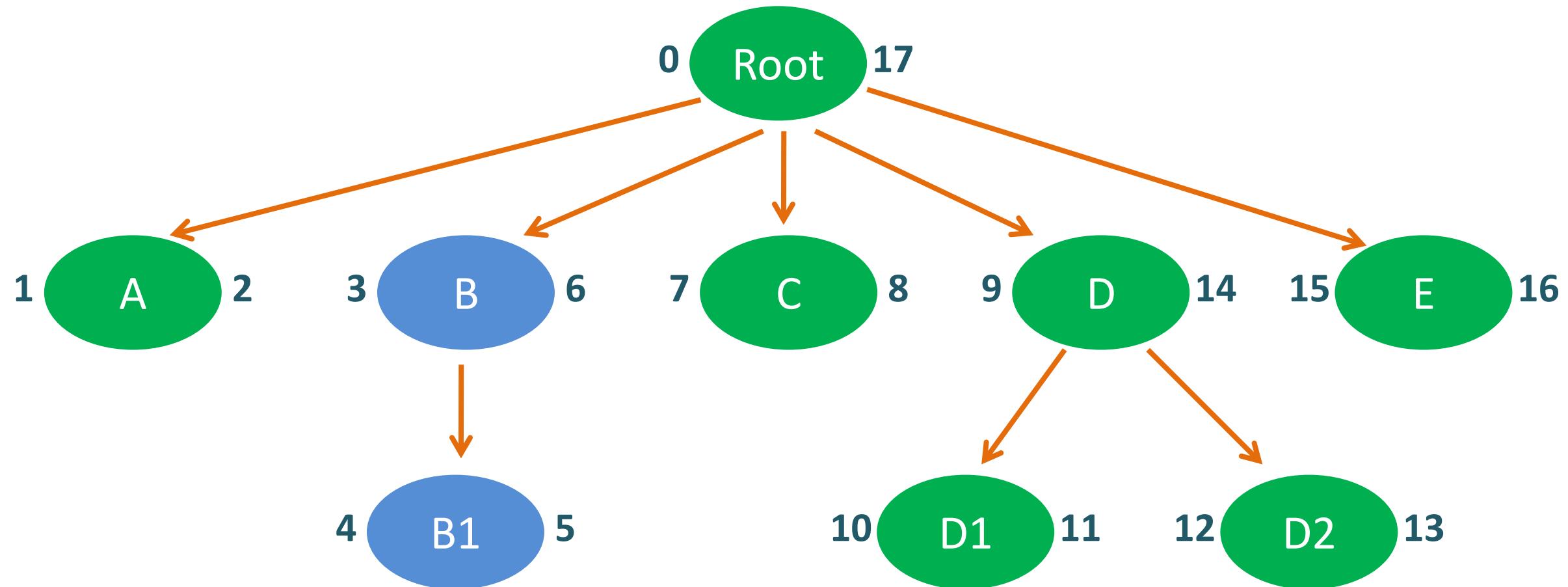
4 trường hợp thêm node vào cây (after)



4 trường hợp thêm node vào cây (after)

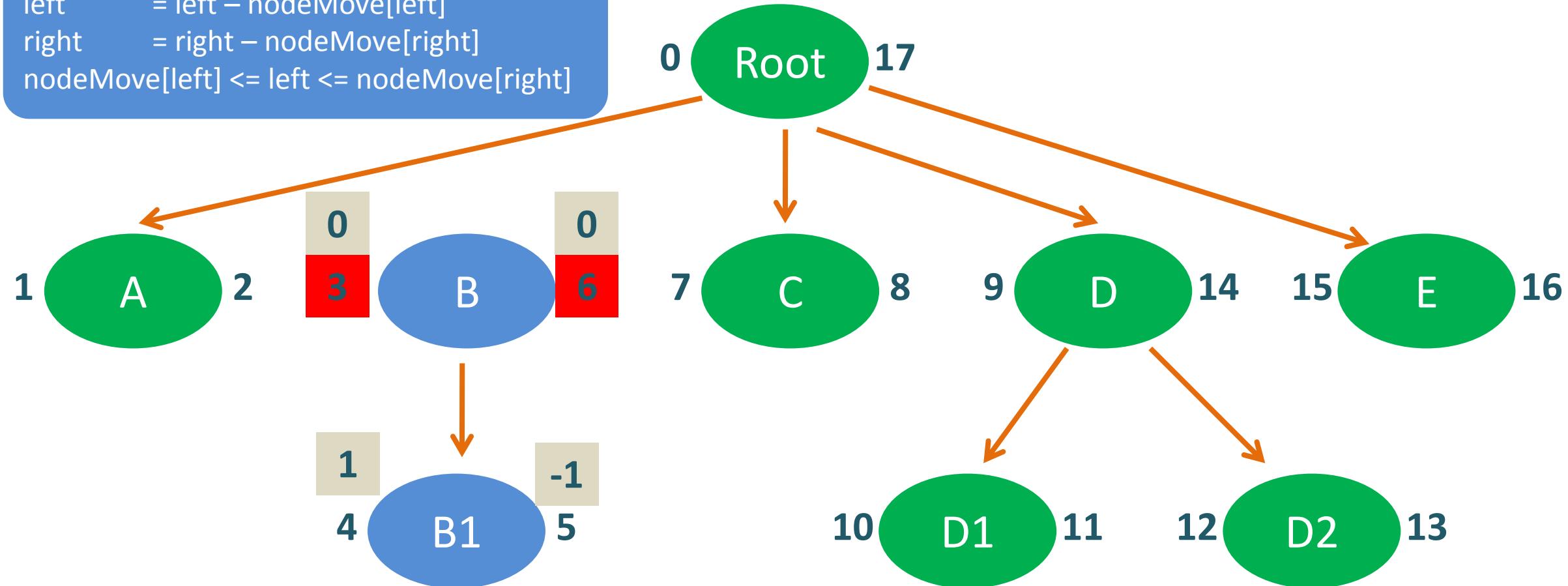


Tách nhánh trên cây (Tình huống)



Tách nhánh trên cây (bước 1)

```
left      = left - nodeMove[left]  
right     = right - nodeMove[right]  
nodeMove[left] <= left <= nodeMove[right]
```



Tách nhánh trên cây (bước 2)

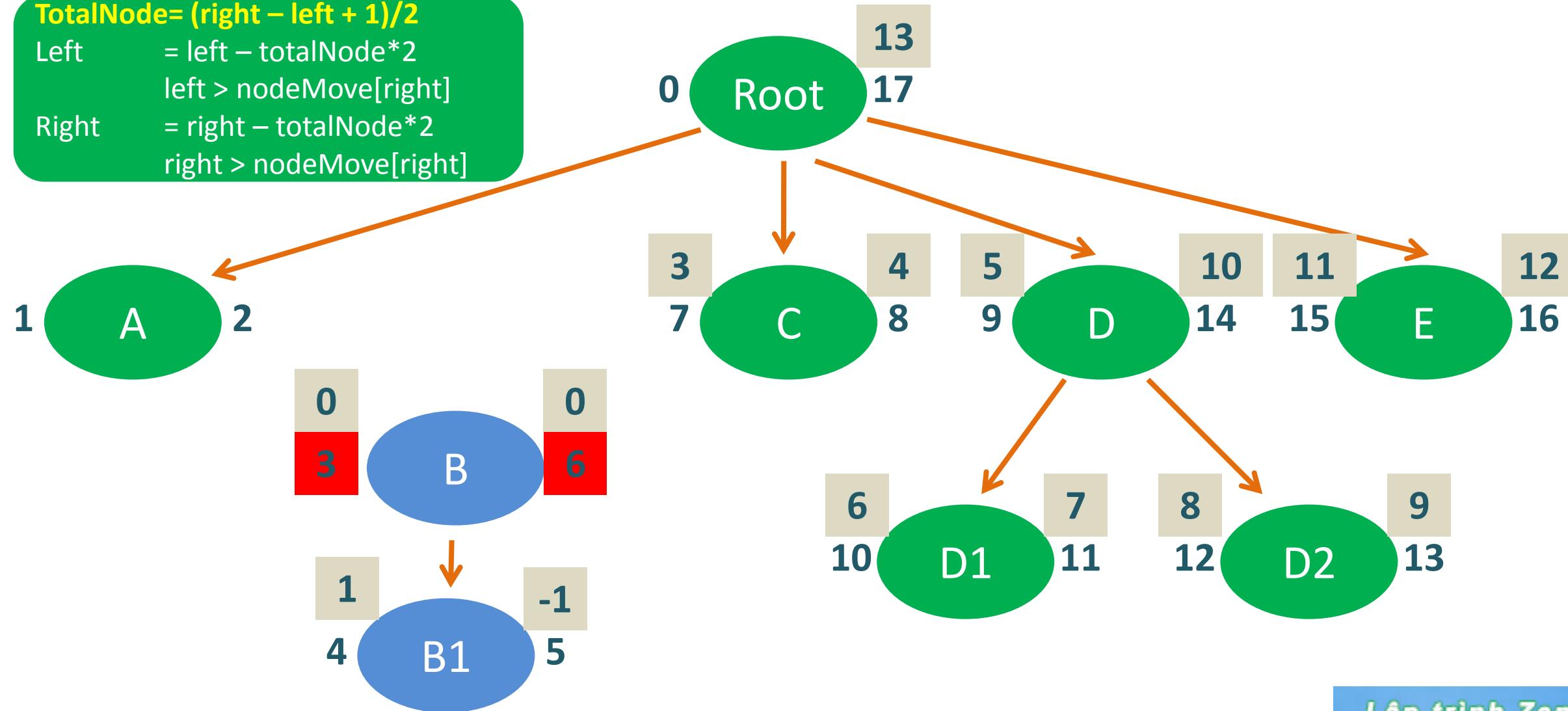
TotalNode= $(right - left + 1)/2$

Left = left - totalNode*2

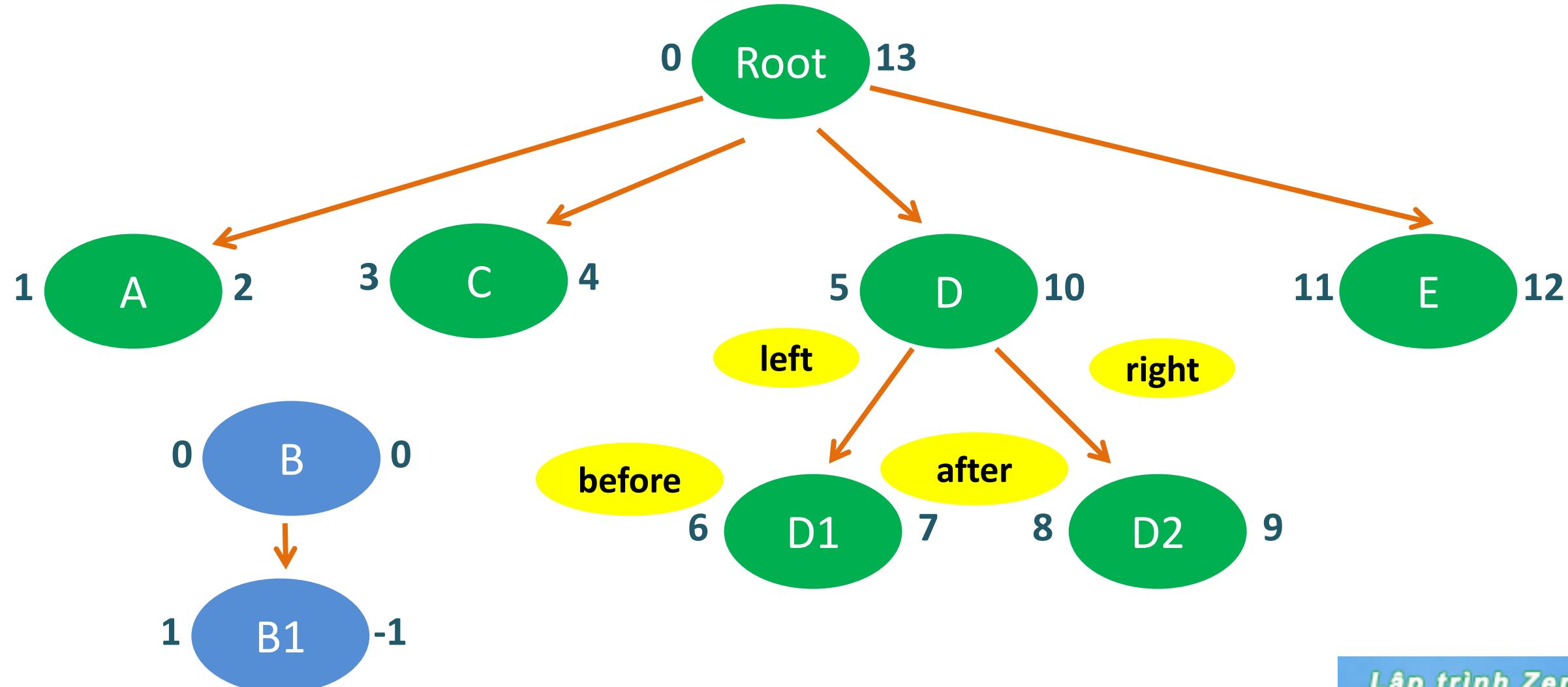
left > nodeMove[right]

Right = right - totalNode*2

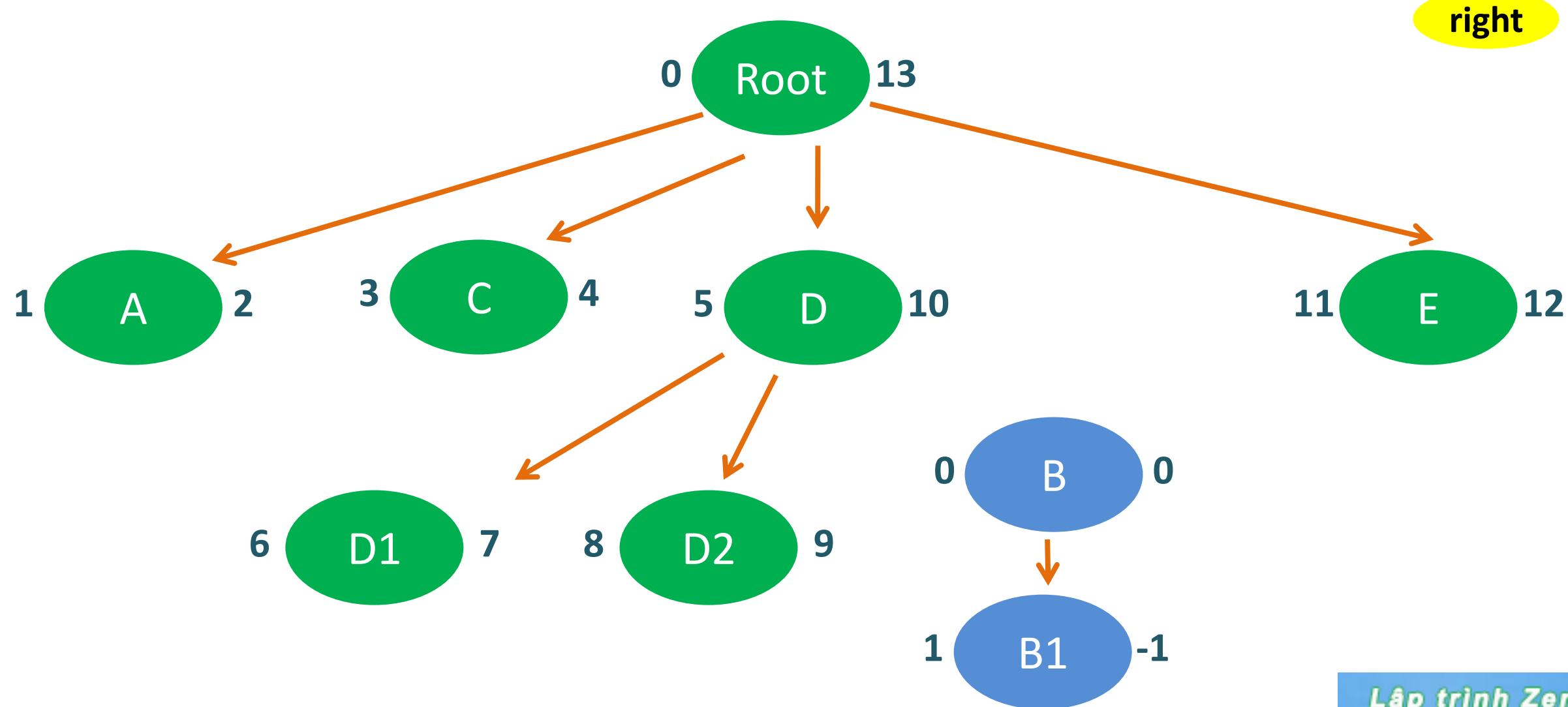
right > nodeMove[right]



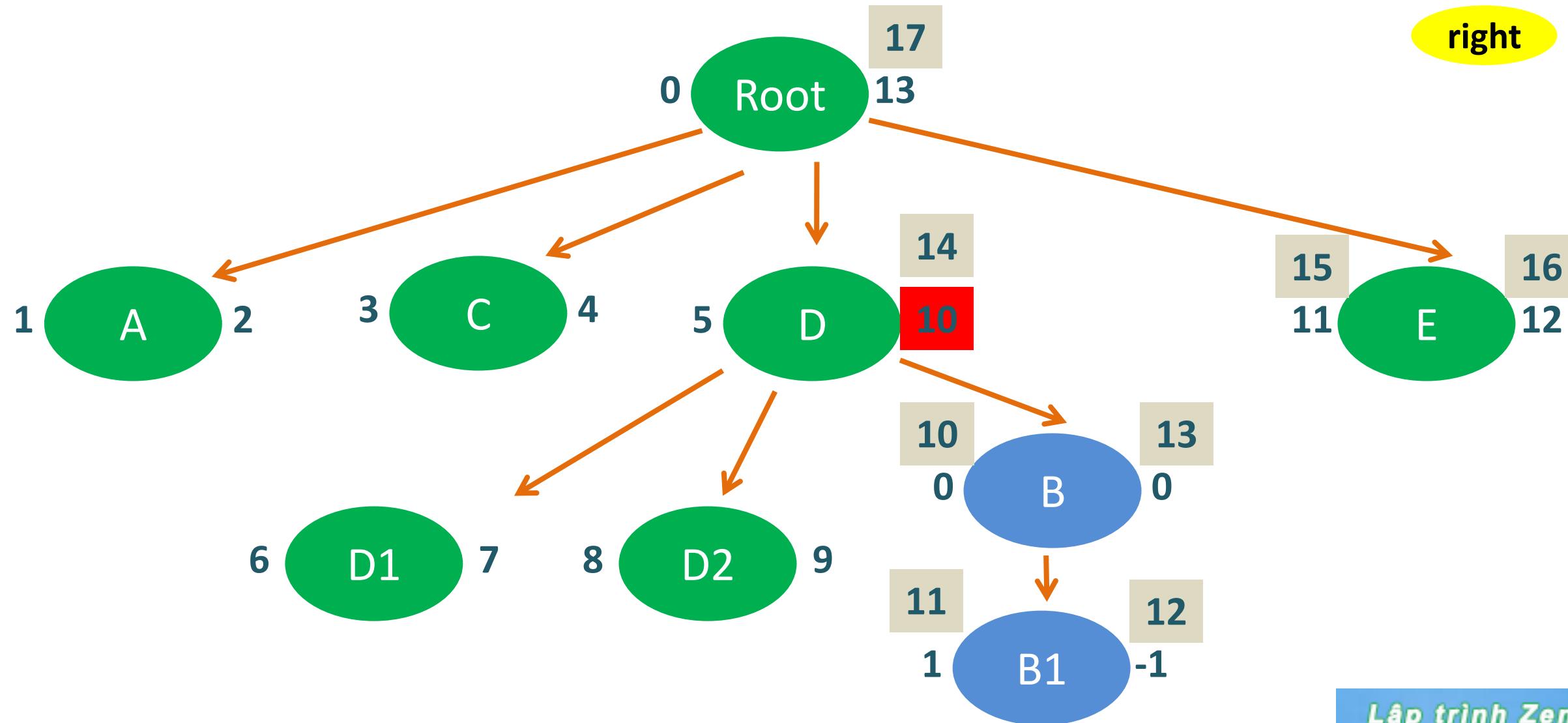
Kết quả tách nhánh trên cây



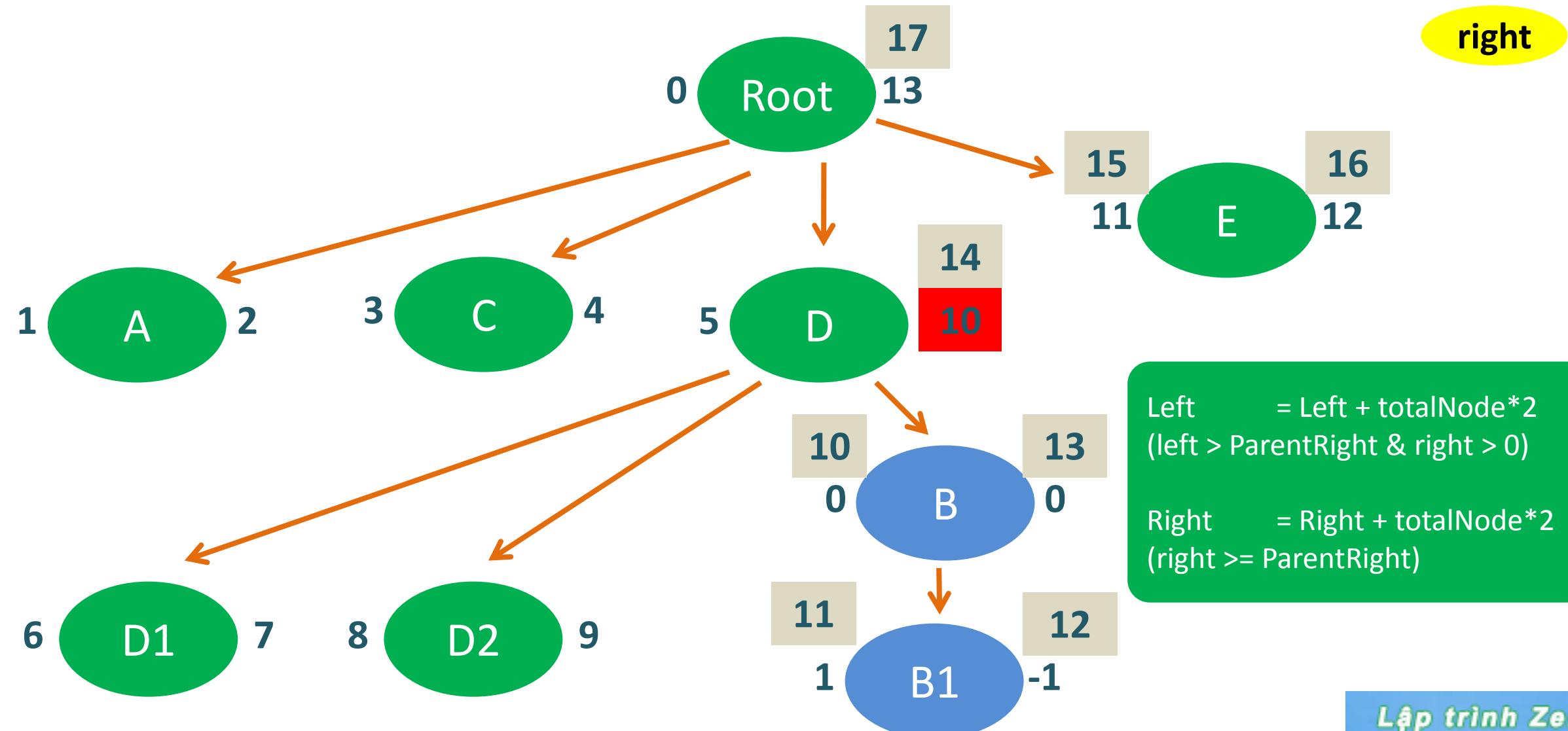
Thêm nhánh vào cây ở vị trí right (tình huống)



Thêm nhánh vào cây ở vị trí right (kết quả)



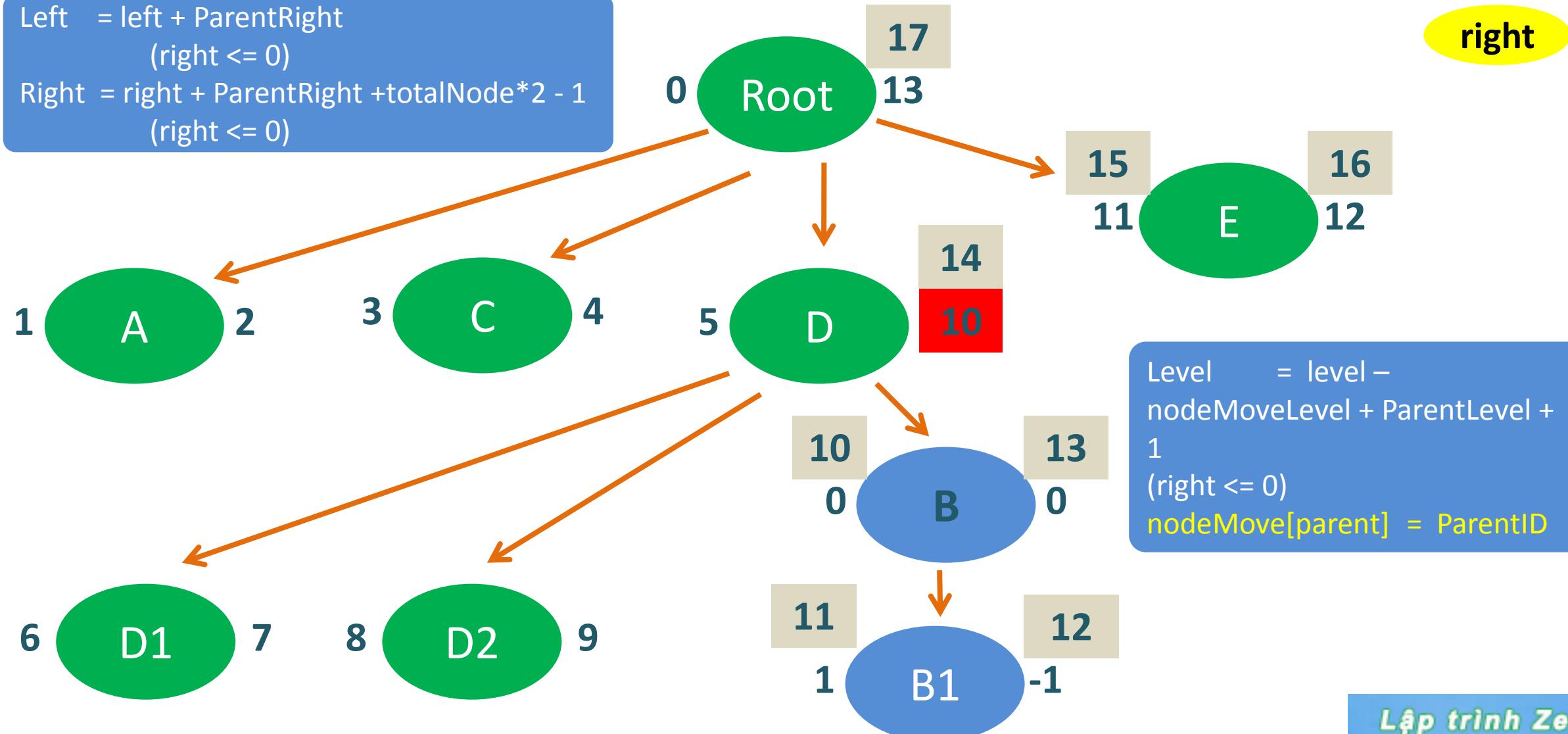
Thêm nhánh vào cây ở vị trí right (thuật toán node trên cây)



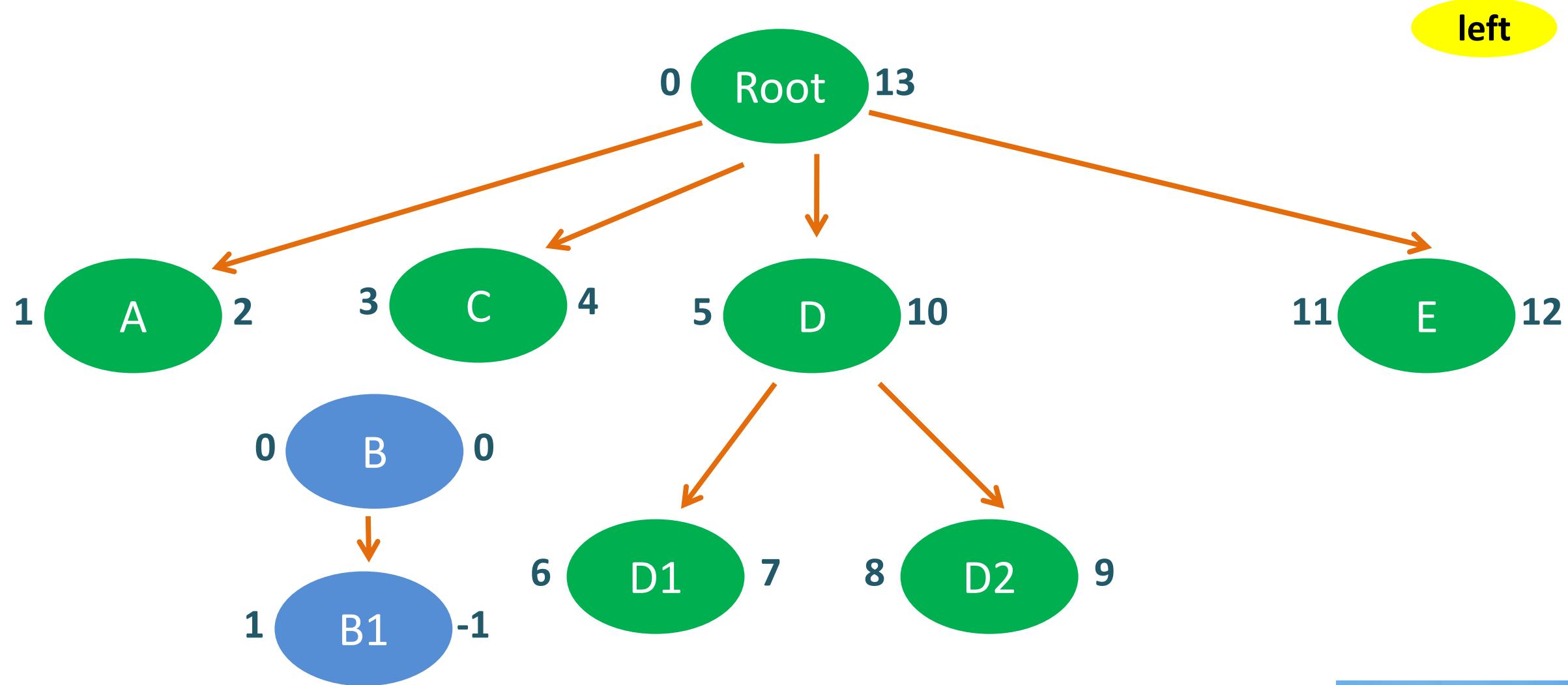
Thêm nhánh vào cây ở vị trí right (thuật toán node trên nhánh)

Left = left + ParentRight
(right <= 0)

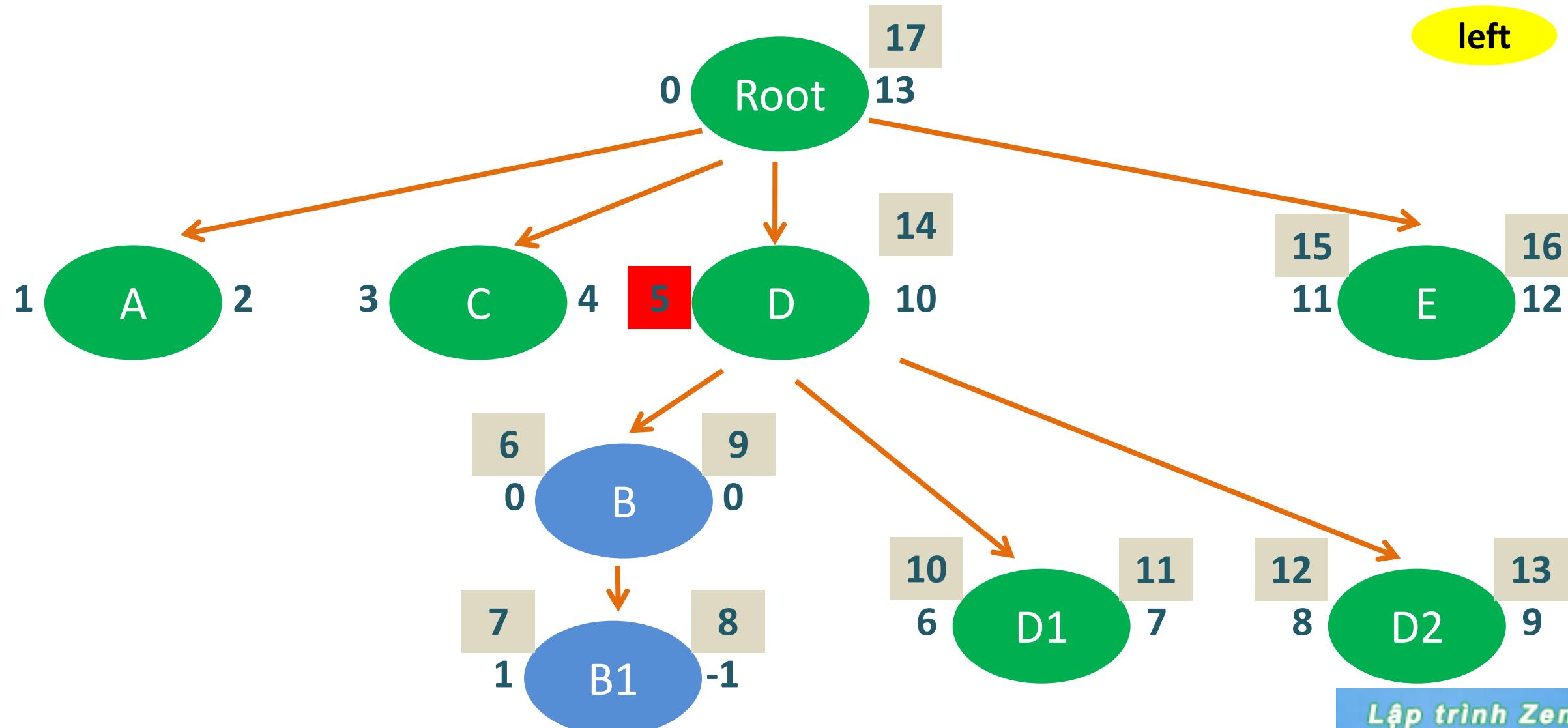
Right = right + ParentRight + totalNode*2 - 1
(right <= 0)



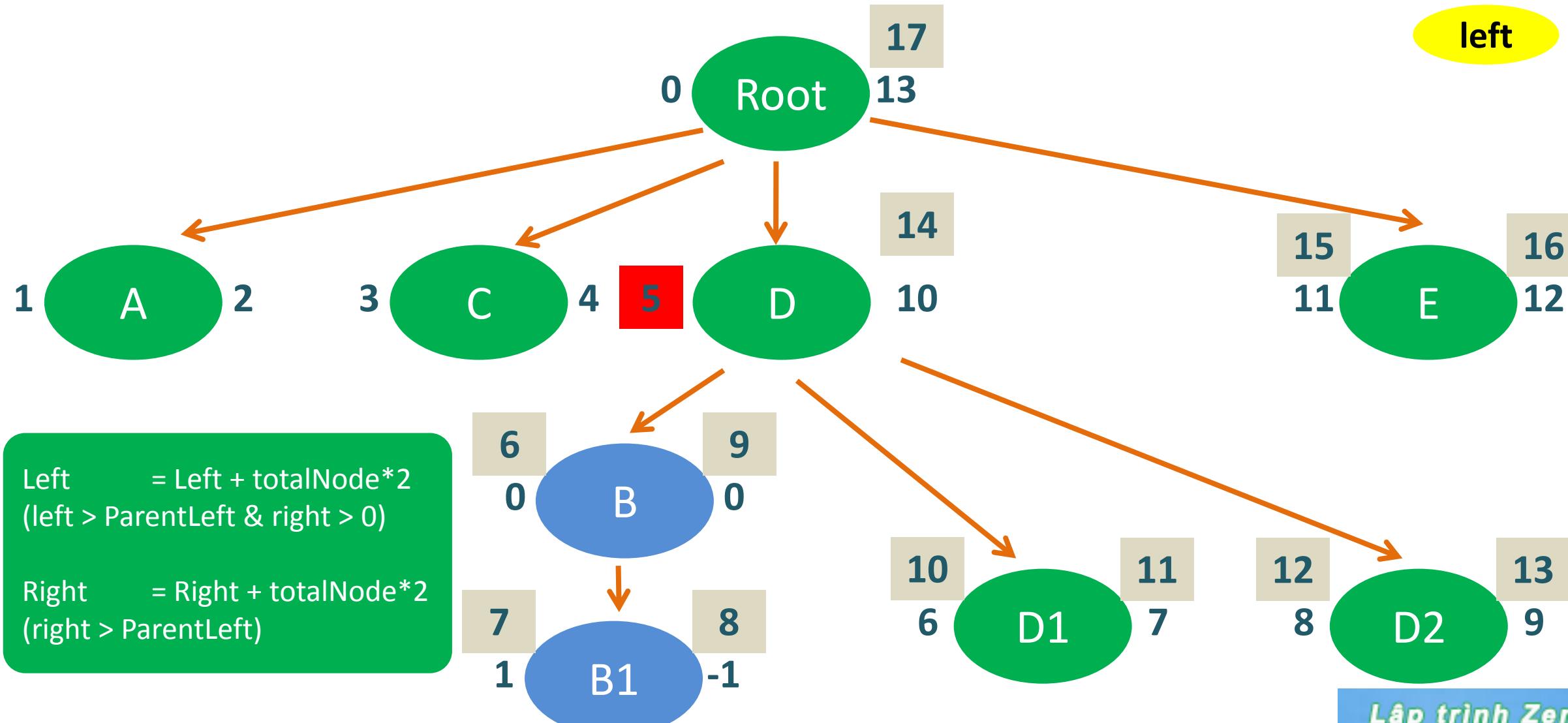
Thêm nhánh vào cây ở vị trí left (tình huống)



Thêm nhánh vào cây ở vị trí left (kết quả)



Thêm nhánh vào cây ở vị trí left (thuật toán node trên cây)



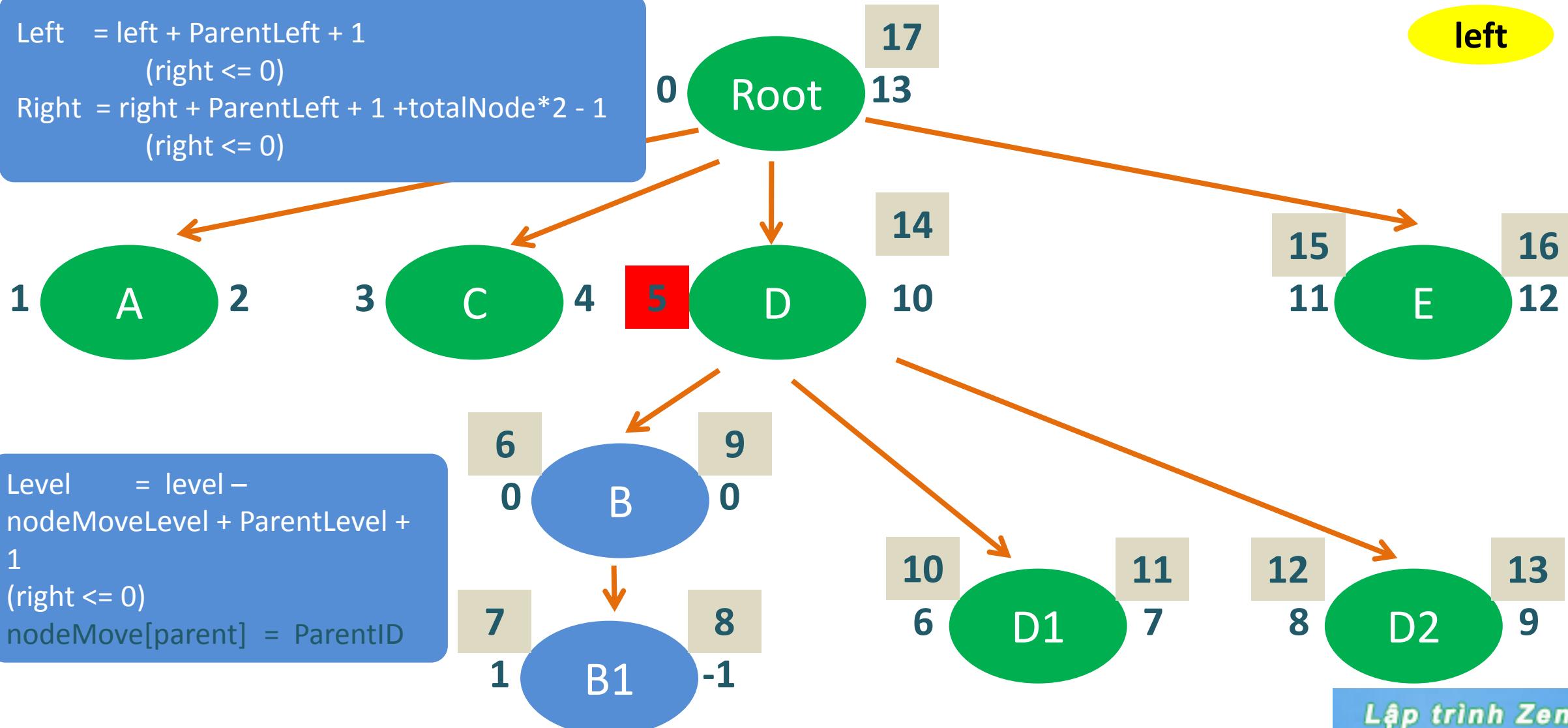
Thêm nhánh vào cây ở vị trí left (thuật toán node trên nhánh)

Left = left + ParentLeft + 1

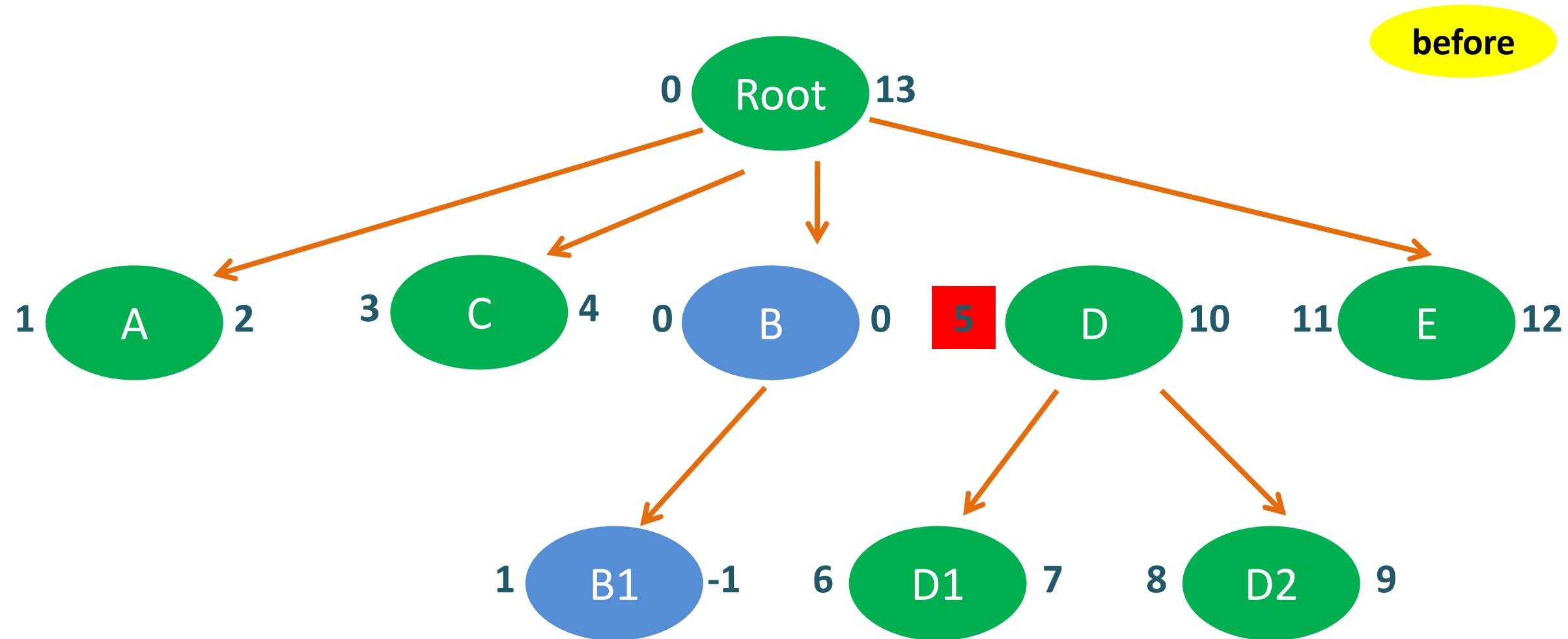
(right <= 0)

Right = right + ParentLeft + 1 + totalNode*2 - 1

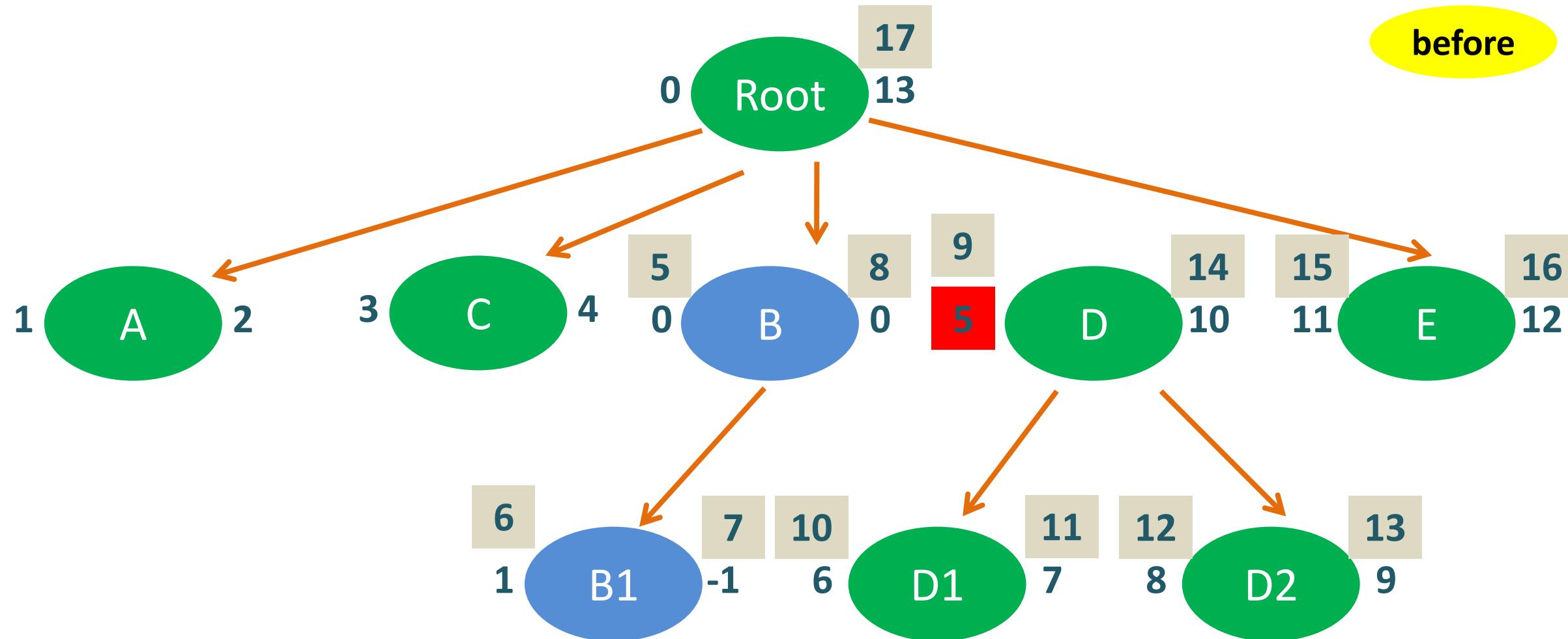
(right <= 0)



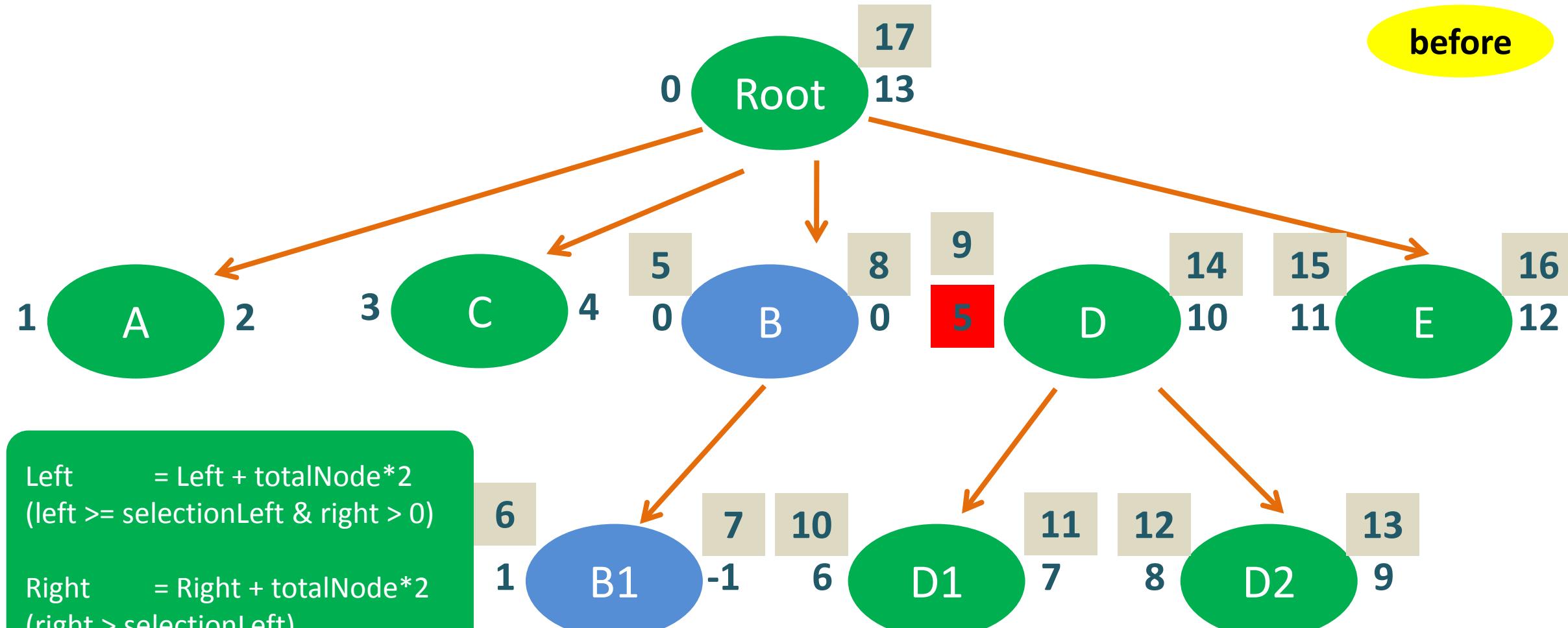
Thêm nhánh vào cây ở vị trí before (tình huống)



Thêm nhánh vào cây ở vị trí before (kết quả)



Thêm nhánh vào cây ở vị trí before (thuật toán node trên cây)



Thêm nhánh vào cây ở vị trí before (thuật toán node trên nhánh)

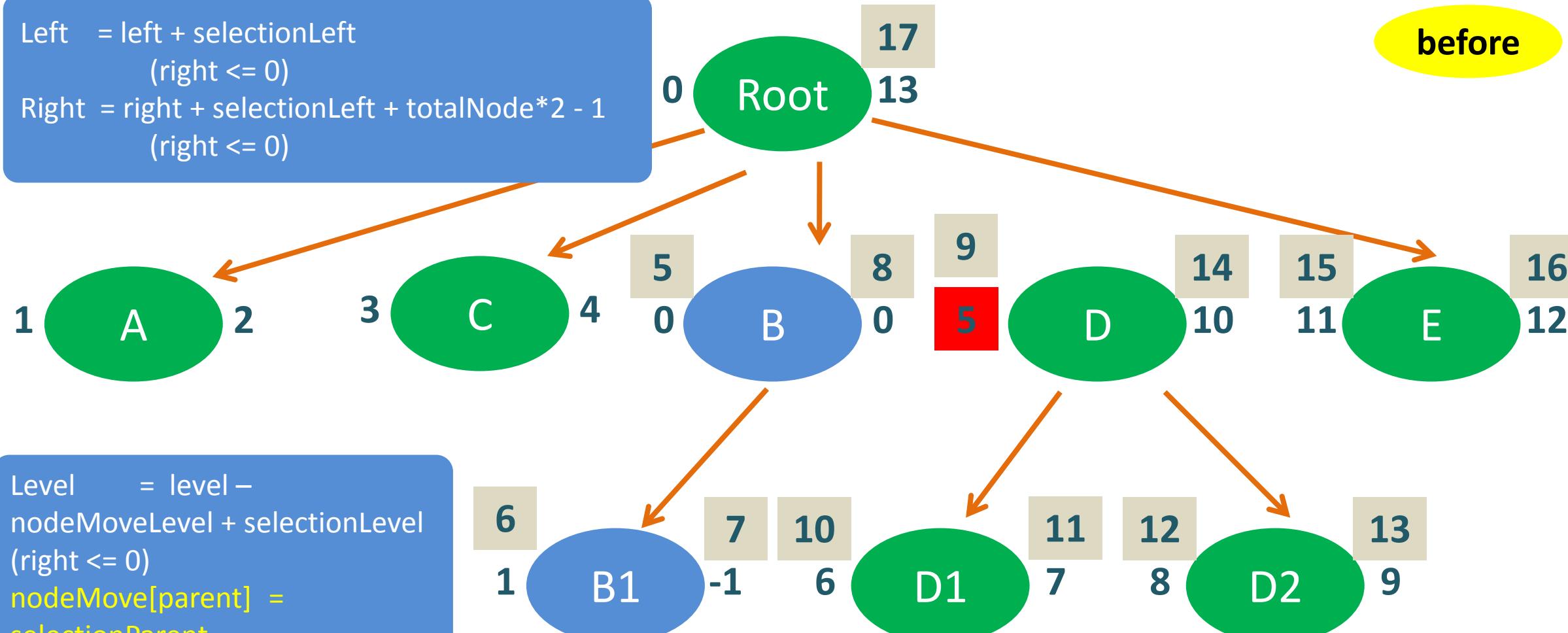
Left = left + selectionLeft

(right <= 0)

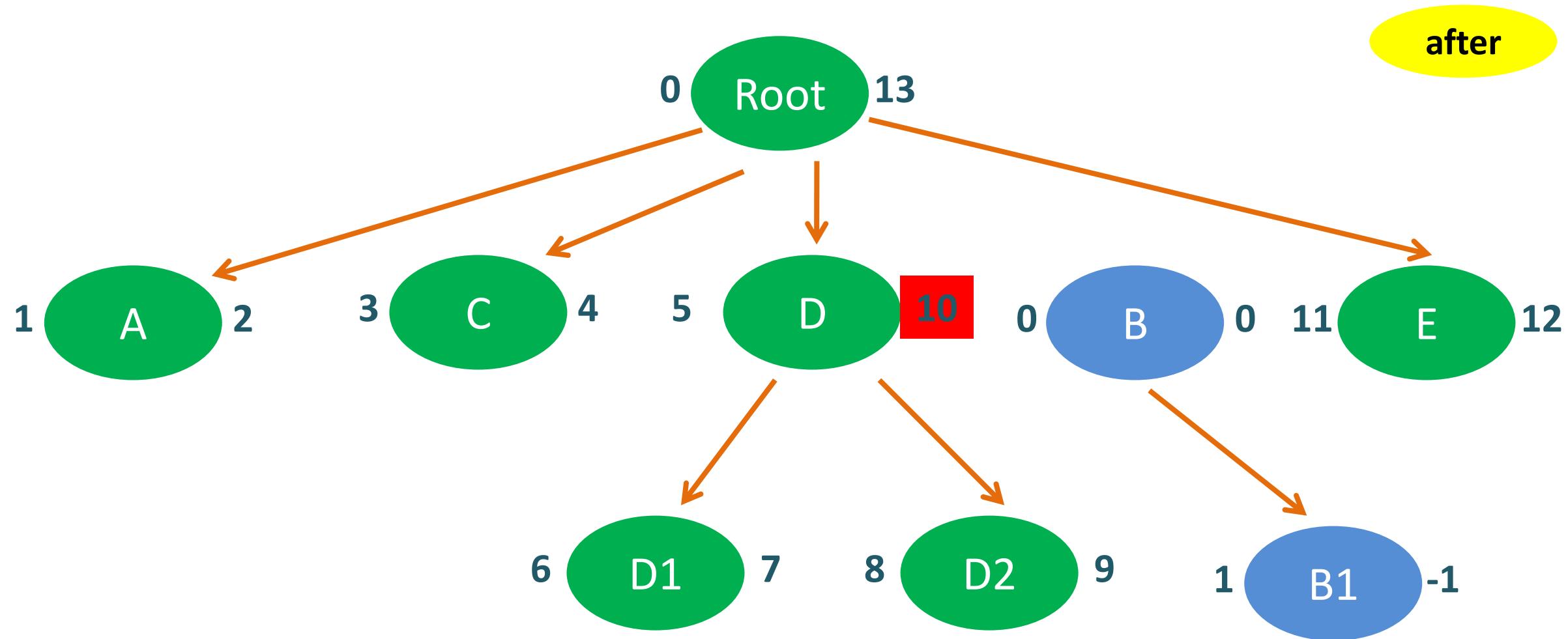
Right = right + selectionLeft + totalNode*2 - 1

(right <= 0)

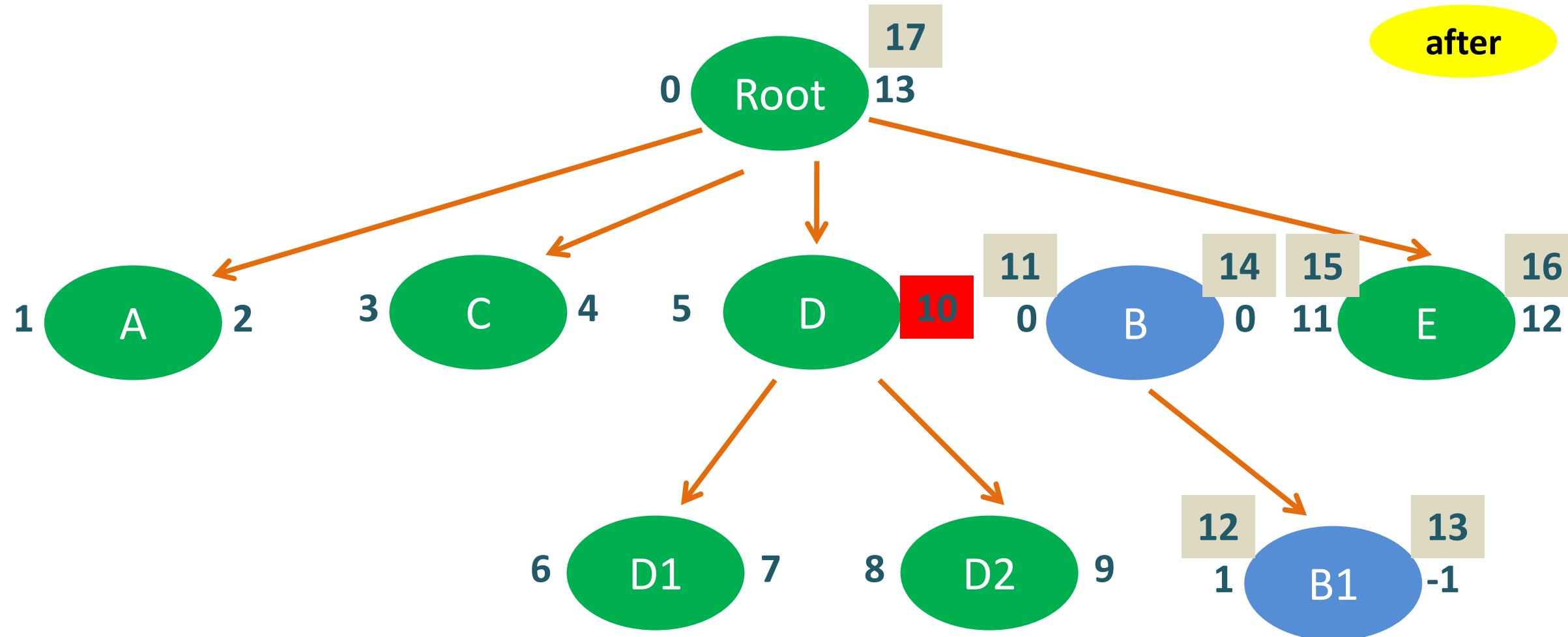
before



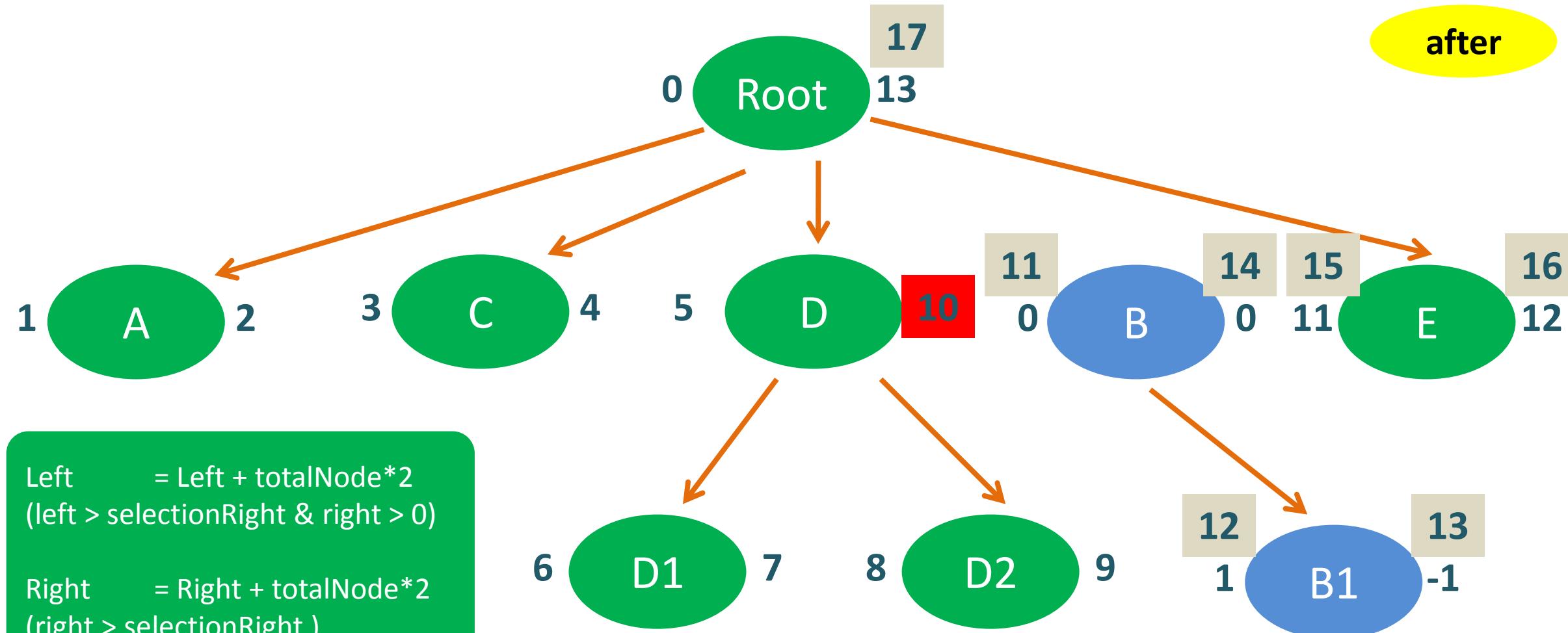
Thêm nhánh vào cây ở vị trí after (tình huống)



Thêm nhánh vào cây ở vị trí after (kết quả)



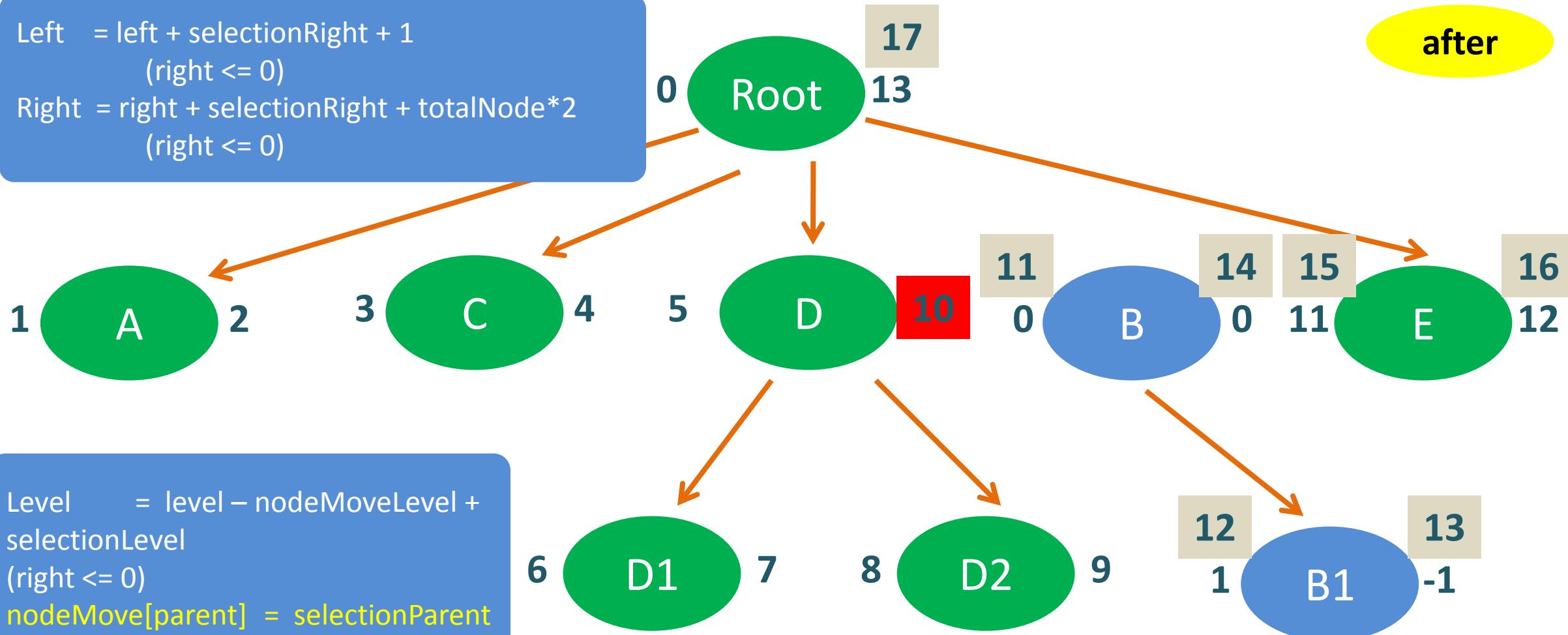
Thêm nhánh vào cây ở vị trí after (thuật toán node trên cây)



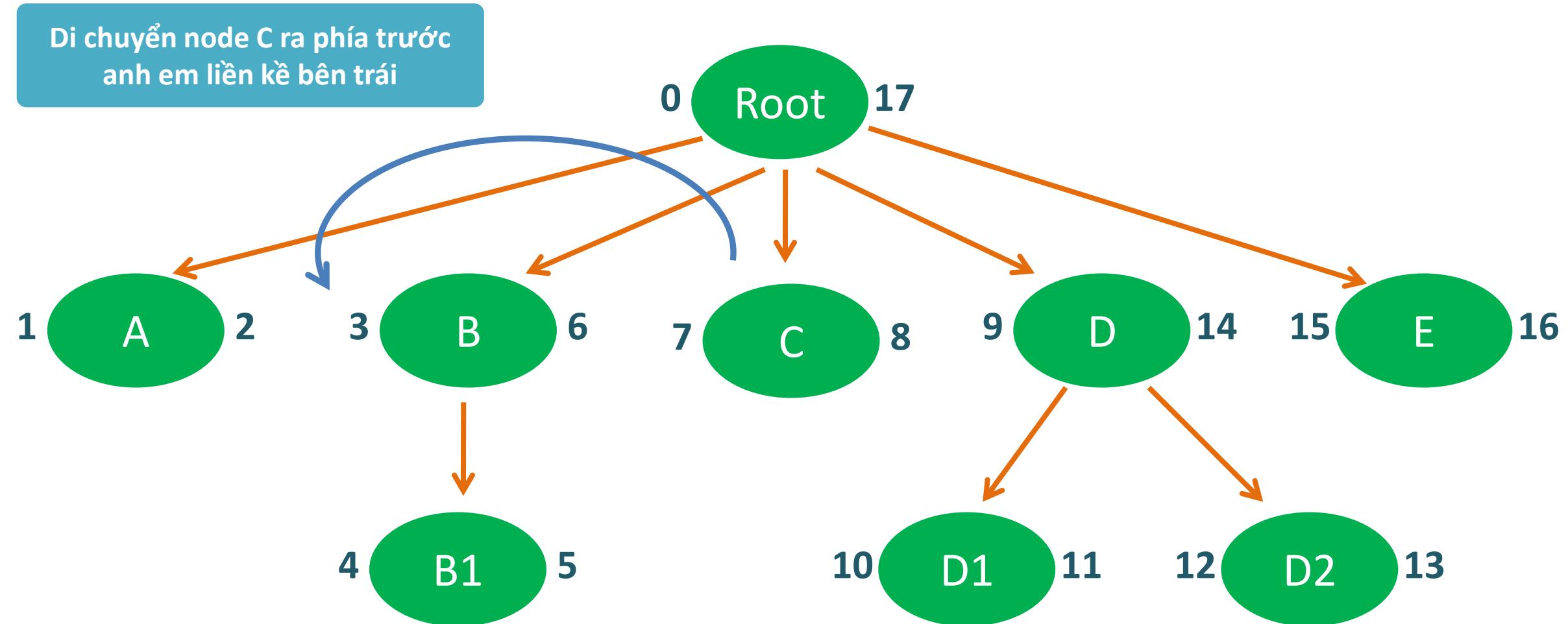
Thêm nhánh vào cây ở vị trí after (thuật toán node trên nhánh)

Left = left + selectionRight + 1
(right <= 0)

Right = right + selectionRight + totalNode*2
(right <= 0)

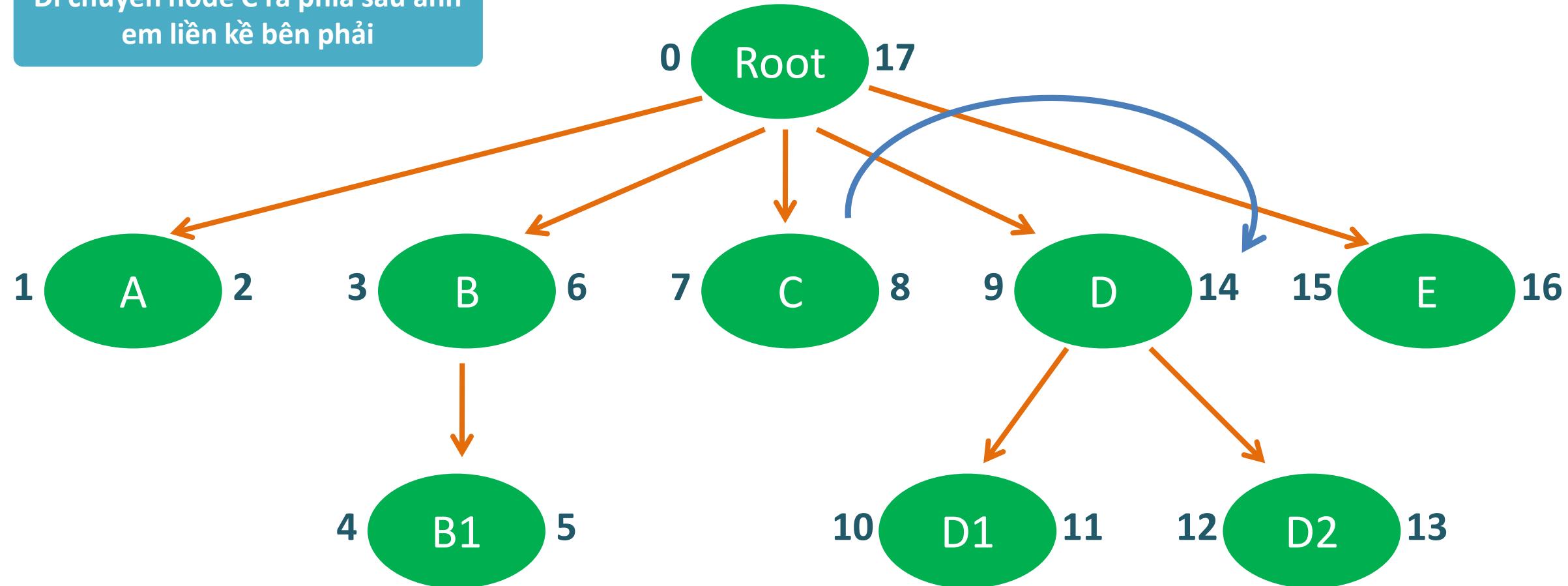


Di chuyển node (move up)



Di chuyển node (move down)

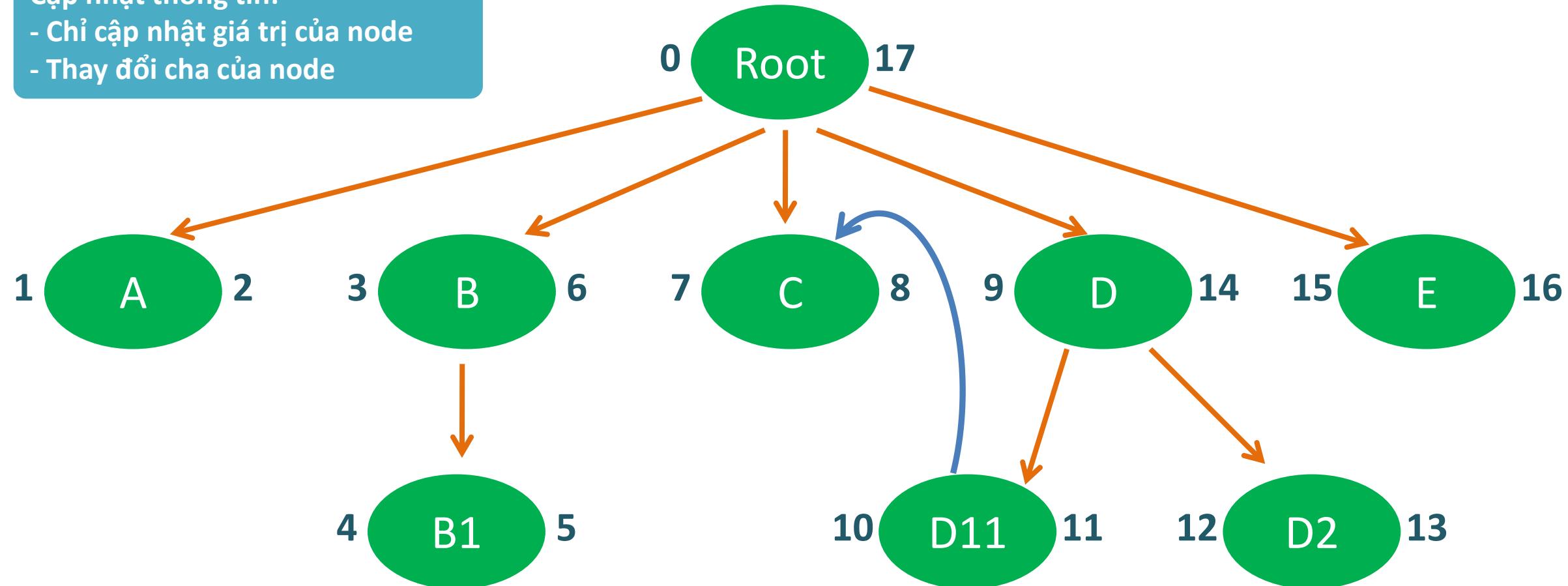
Di chuyển node C ra phía sau anh em liền kề bên phải



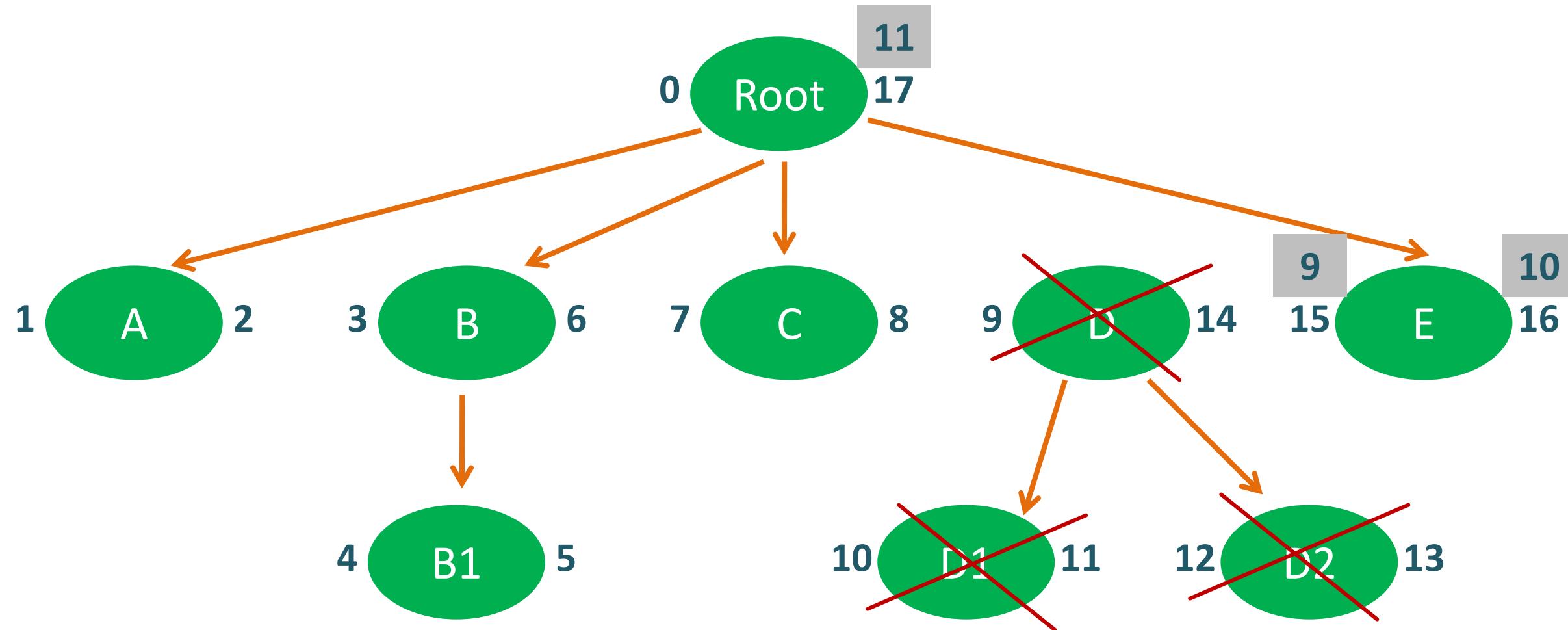
Cập nhật thông tin của node

Cập nhật thông tin:

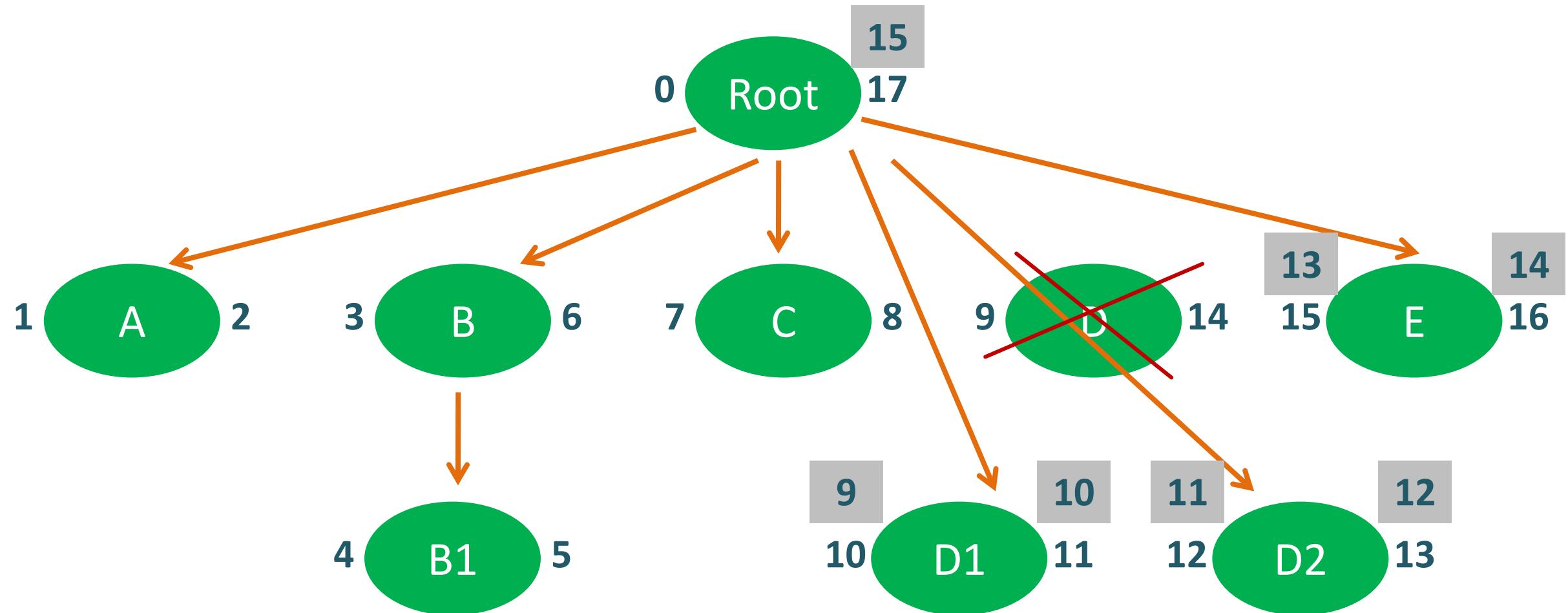
- Chỉ cập nhật giá trị của node
- Thay đổi cha của node



Xóa node trên cây (xóa toàn bộ nhánh)



Xóa node trên cây (chỉ xóa node)



Quản trị danh mục sách theo Nested Set Model

1. Hiển thị danh sách category có phân cấp cha con
2. Hiển thị ordering
3. Sắp xếp và lọc danh sách category: column, status, keyword, level
4. Thay đổi trạng thái của một hoặc nhiều category: inactive, active
5. Xóa một hoặc nhiều category
6. Di chuyển category: move up, move down
7. Cập nhật thông tin của một category: add, edit

Xây dựng chức năng quản lý sách

1. Nhận xét: cách xây dựng chức năng quản lý sách hoàn toàn tương tự như chức năng quản lý user
2. Đối với trang danh sách các quyền sách cần bổ sung: show special, filter special, change special, show price, show sale off
3. Đối với trang thêm mới (cập nhật) thông tin quyền sách cần bổ sung: special, price, sale off, customize description