



TRƯỜNG ĐẠI HỌC CẦN THƠ
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
KHOA CÔNG NGHỆ THÔNG TIN

QUẢN TRỊ DỮ LIỆU - CT467

Chương 2: LƯU TRỮ VÀ
CẤU TRÚC TẬP TIN

Biên soạn:



Ths. Nguyễn Thị Kim Yến



Ntkyen@ctu.edu.vn



CANTHO UNIVERSITY

NỘI DUNG

1

Các loại thiết bị lưu trữ

2

Đĩa từ và RAID

3

Quản lý bộ đệm

4

Tổ chức tập tin

5

Tổ chức mẫu tin trong tập tin

6

Chỉ mục và băm

7

Xử lý vấn tin



MỤC TIÊU

Nhằm giới thiệu 2 kỹ thuật rất phổ biến được sử dụng để tăng hiệu suất truy cập dữ liệu trong các Hệ QT CSDL là chỉ mục (index) và băm (hash)



6. Chỉ mục và băm

Chỉ mục

Chỉ mục
được sắp

Chỉ mục
B⁺-cây

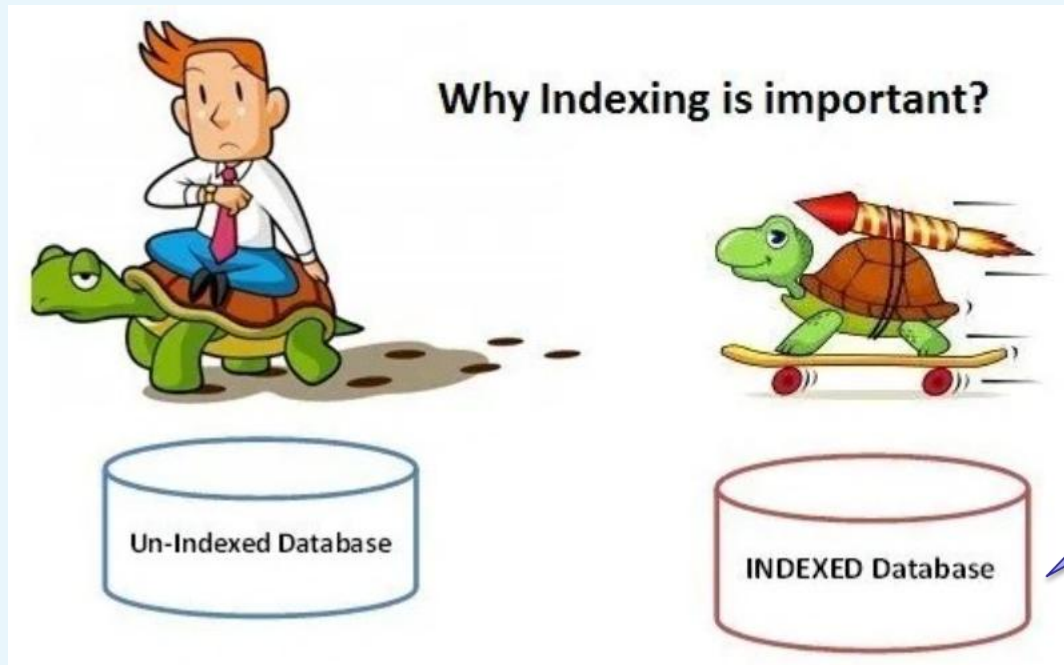
Băm

Băm tĩnh

Băm
động

6.1 Chỉ mục (indexing)

- **Chỉ mục:** một cấu trúc dữ liệu được dùng để định vị và truy cập nhanh nhất vào trong các bảng DB
- Một Index là một con trỏ tới dữ liệu trong một bảng



Tăng tốc độ tìm kiếm
và truy cập DB



6.1 Chỉ mục (indexing) (tt)

- **Cấu trúc của Index:**

Search Key	Data Reference
------------	----------------

Structure of an index

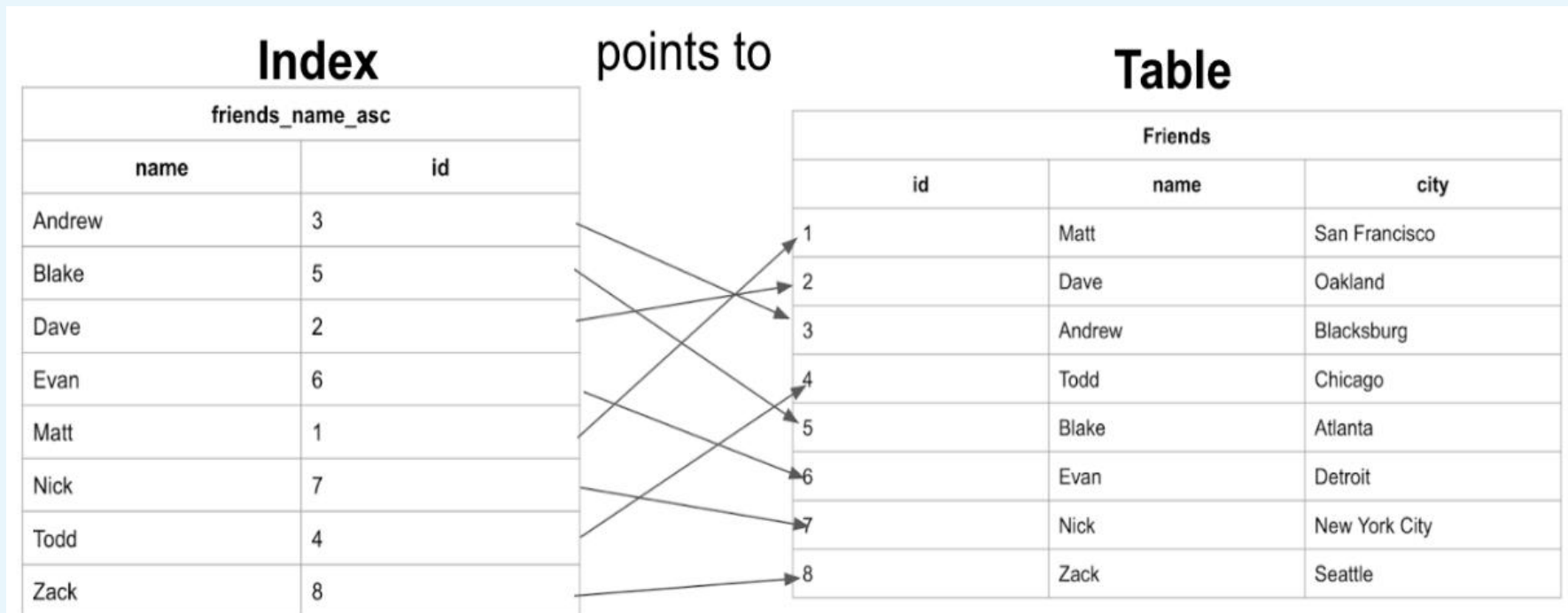
Chứa bản sao các giá trị của cột được tạo Index

Chứa con trỏ trỏ đến địa chỉ của bản ghi có giá trị cột index tương ứng



6.1.1 Chỉ mục được sắp (Ordered index)

- Là chỉ mục lưu trữ các giá trị khoá tìm kiếm **trong thứ tự được sắp** và kết hợp từng khoá tìm kiếm với các mẫu tin chứa khoá tìm kiếm này





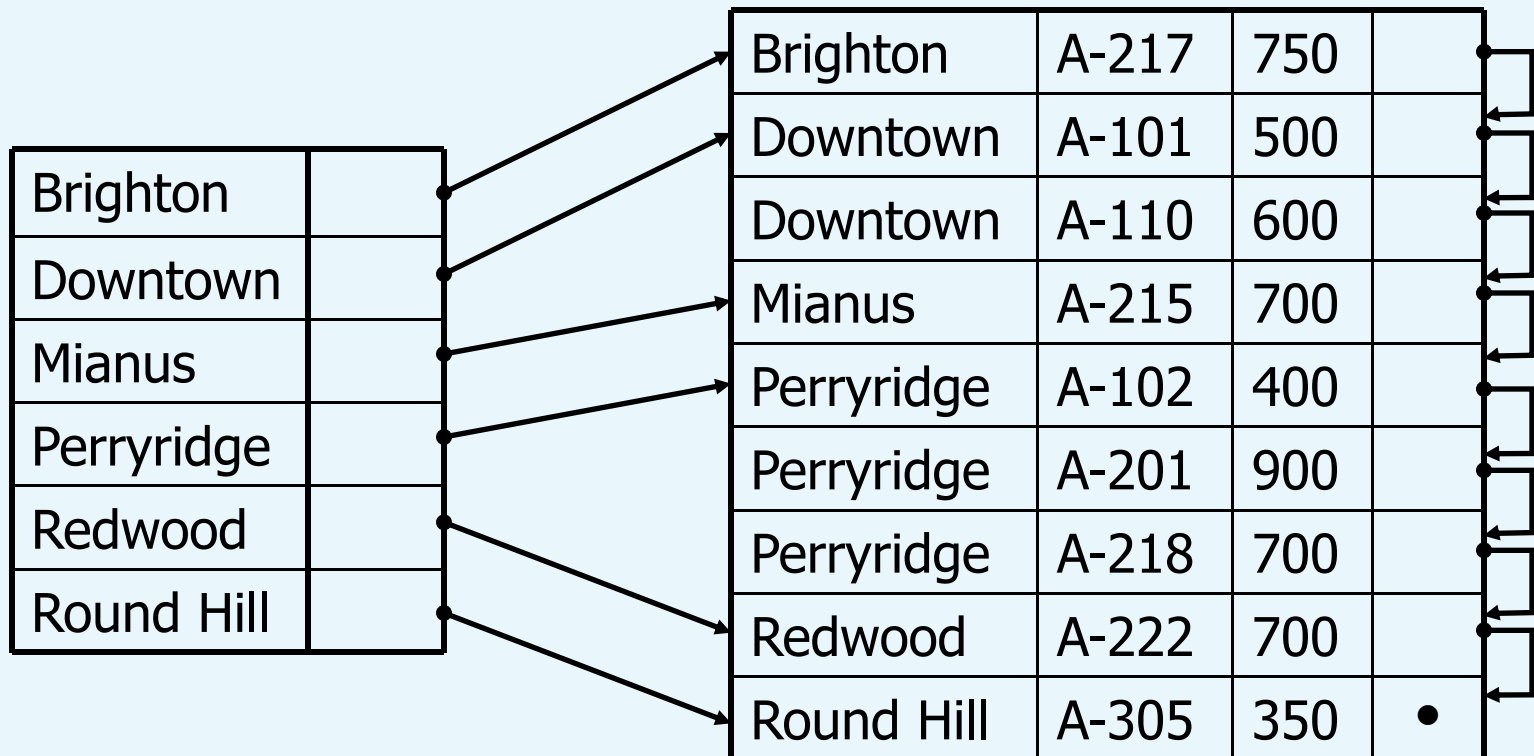
6.1.1 Chỉ mục được sắp (Ordered index) (tt)

- Nếu file chứa các mẫu tin được sắp tuần tự theo khóa tìm kiếm, chỉ mục theo khoá tìm kiếm này được gọi là **chỉ mục sơ cấp** (hay chỉ mục theo cụm)
- Nếu file chứa các mẫu tin không được sắp theo khóa tìm kiếm, chỉ mục theo khóa tìm kiếm này được gọi là **chỉ mục thứ cấp** (hay chỉ mục không phân cụm)



a) Chỉ mục sơ cấp (Primary index)

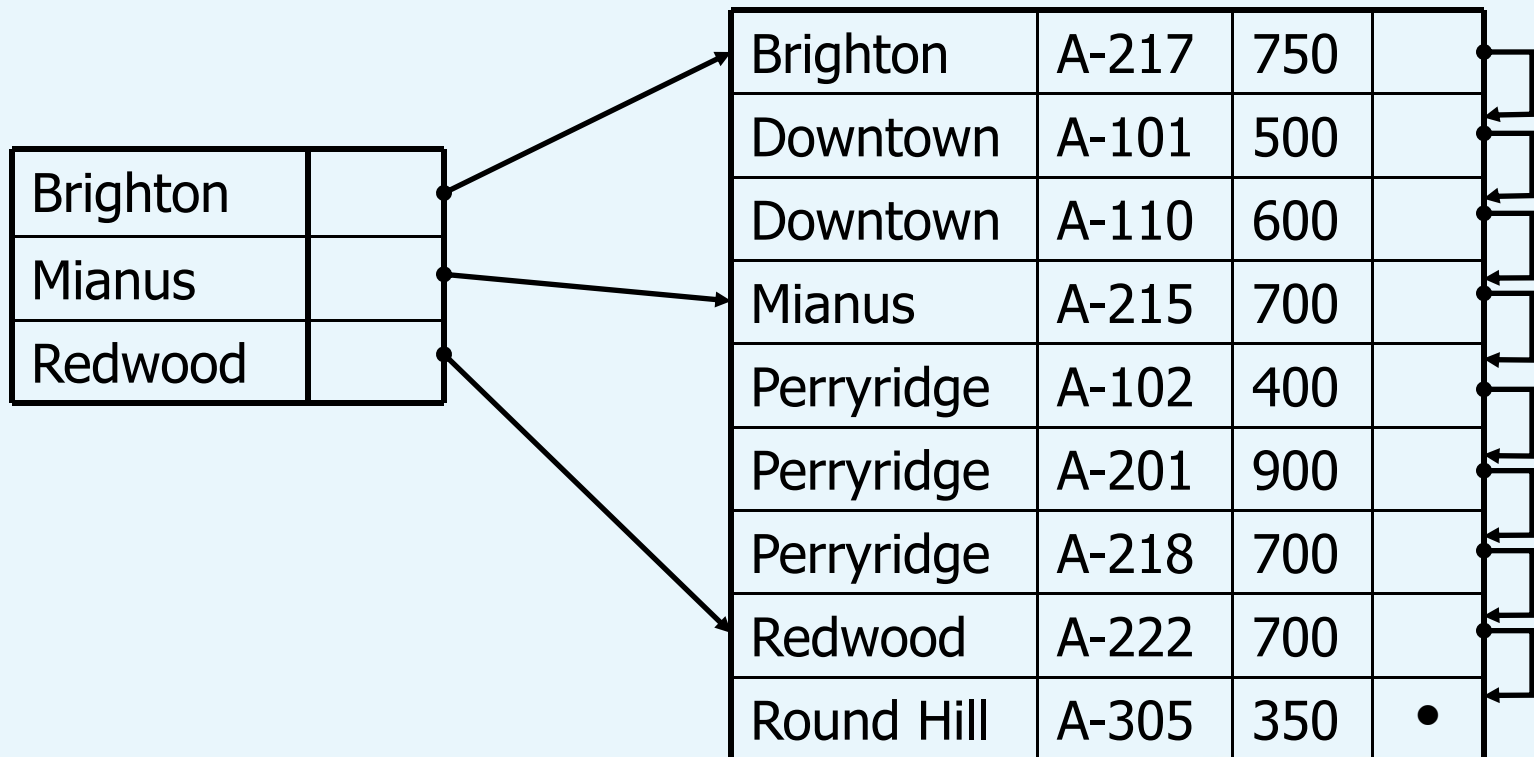
- **Chỉ mục đặc (dense index):** có mẫu tin chỉ mục cho mỗi khóa tìm kiếm





a) Chỉ mục sơ cấp (Primary index) (tt)

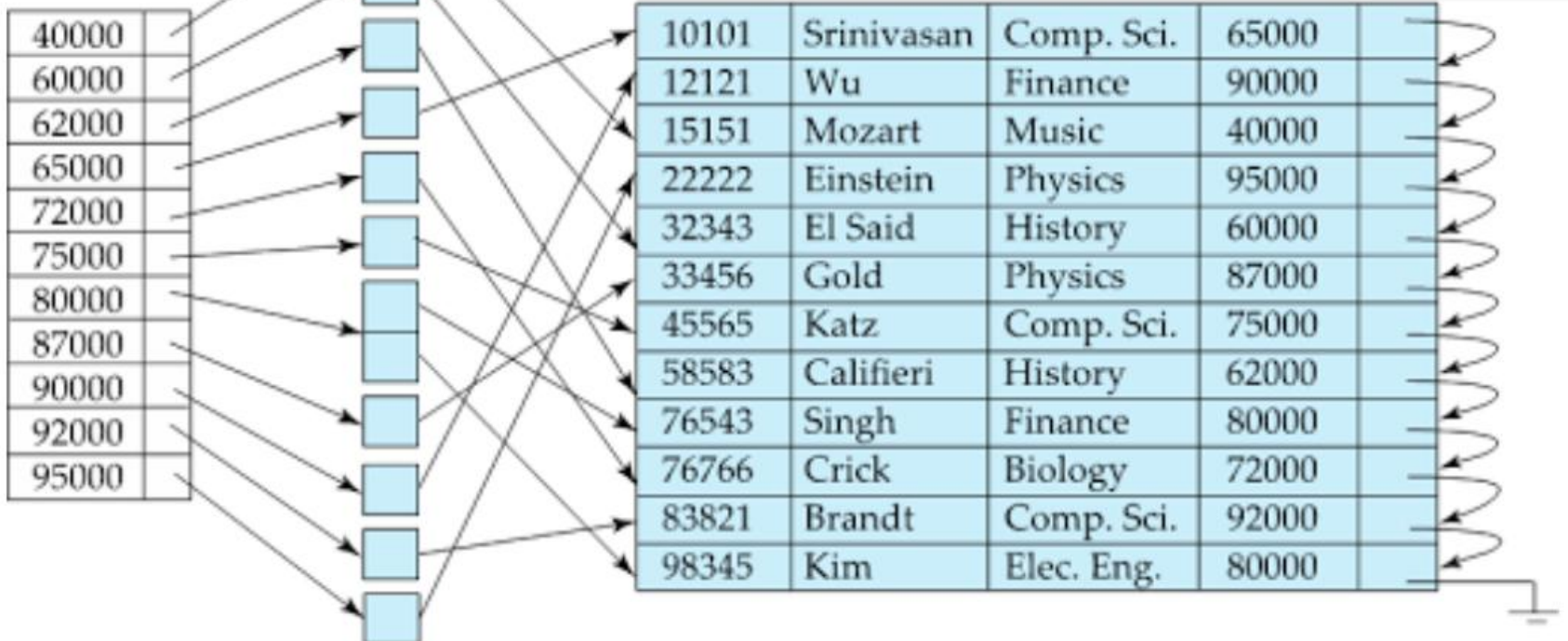
- Chỉ mục thưa (sparse index): chỉ có mẫu tin chỉ mục cho một số khóa tìm kiếm



b) Chỉ mục thứ cấp (Secondary index)

- Mỗi index record trỏ tới một bucket chứa các con trỏ tới tất cả các bản ghi có giá trị search-key.

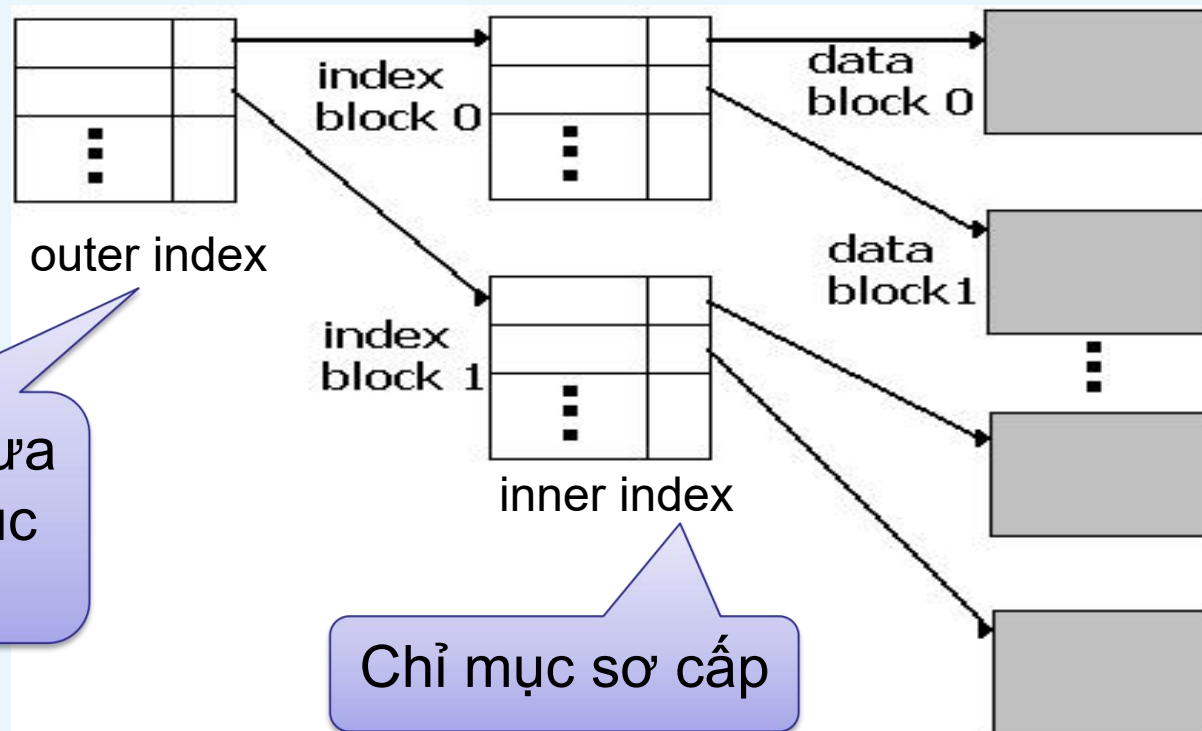
Bucket: 1 đơn vị lưu trữ chứa 1 hoặc nhiều record





Chỉ mục nhiều cấp (Multilevel index)

- Sử dụng trong trường hợp tập tin chỉ mục quá lớn
- Cách thực hiện:** Lưu chỉ mục sơ cấp trên ổ cứng và xây một chỉ mục thừa trên nó.

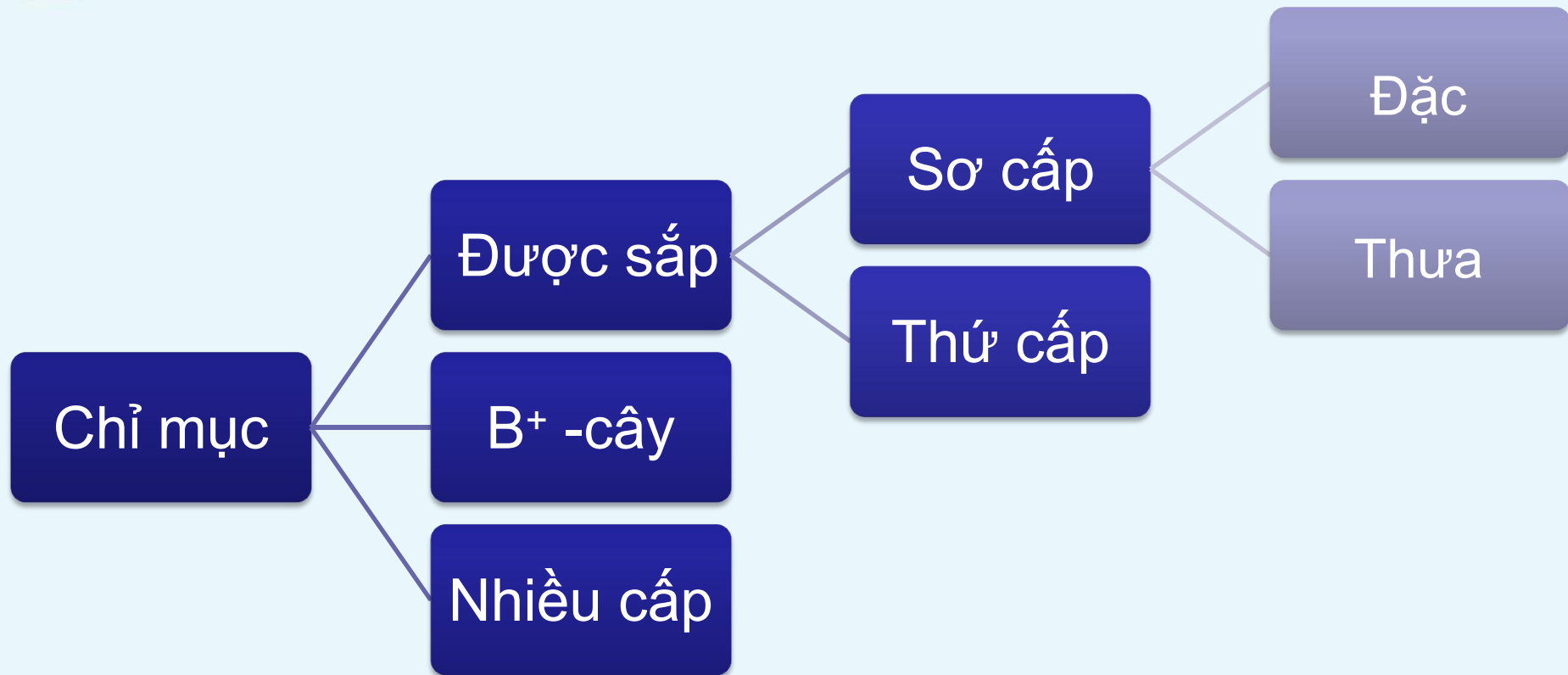


Chỉ mục thừa
của chỉ mục
sơ cấp

Chỉ mục sơ cấp



6.1 Chỉ mục (indexing) (tt)





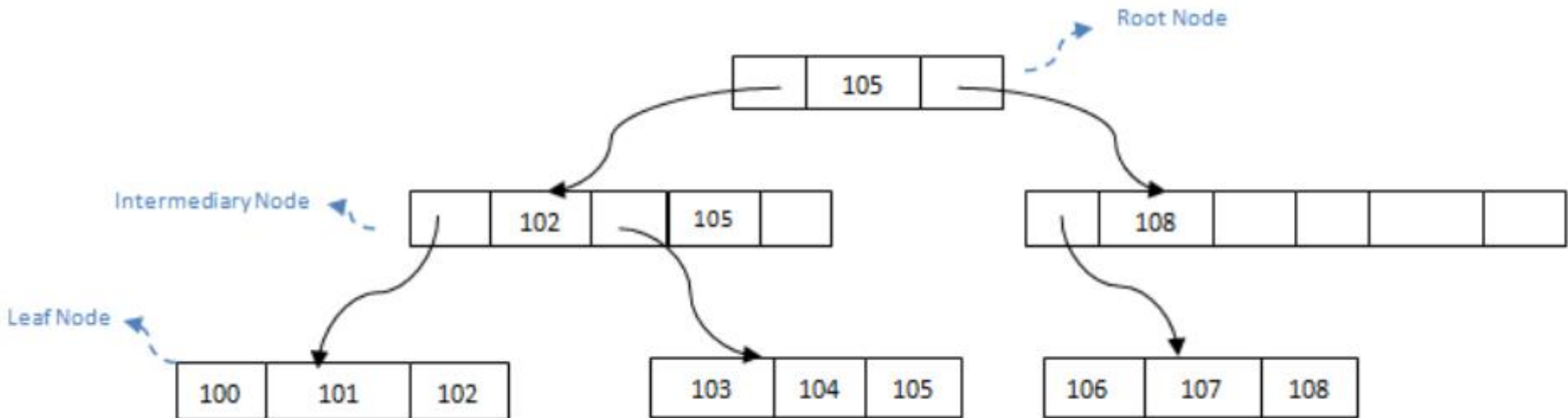
6.1.2 File chỉ mục B⁺-Cây (B⁺-Tree Index file)

- **Nhược điểm** của file chỉ mục tuần tự: **giảm hiệu năng** khi file lớn lên
 - **Đề khắc phục**: đòi hỏi phải tổ chức lại file
- **B⁺-cây** là một cấu trúc tổ chức chỉ mục được sử dụng rộng rãi vì nó **đảm bảo được tính hiệu quả** bất chấp các hành động xen xoá.

6.1.2 File chỉ mục B⁺ -Cây (tt)

STUDENT		
STUDENT ID	STUDENT NAME	ADDRESS
100	Joseph	Alaiedon Township
101	Allen	Fraser Township
102	Chris	Clinton Township
103	Patty	Troy
104	Jack	Fraser Township
105	Jessica	Clinton Township
106	James	Troy
107	Antony	Alaiedon Township
108	Jacob	Troy

Nút bên phải sẽ có giá trị **lớn hơn nút gốc**



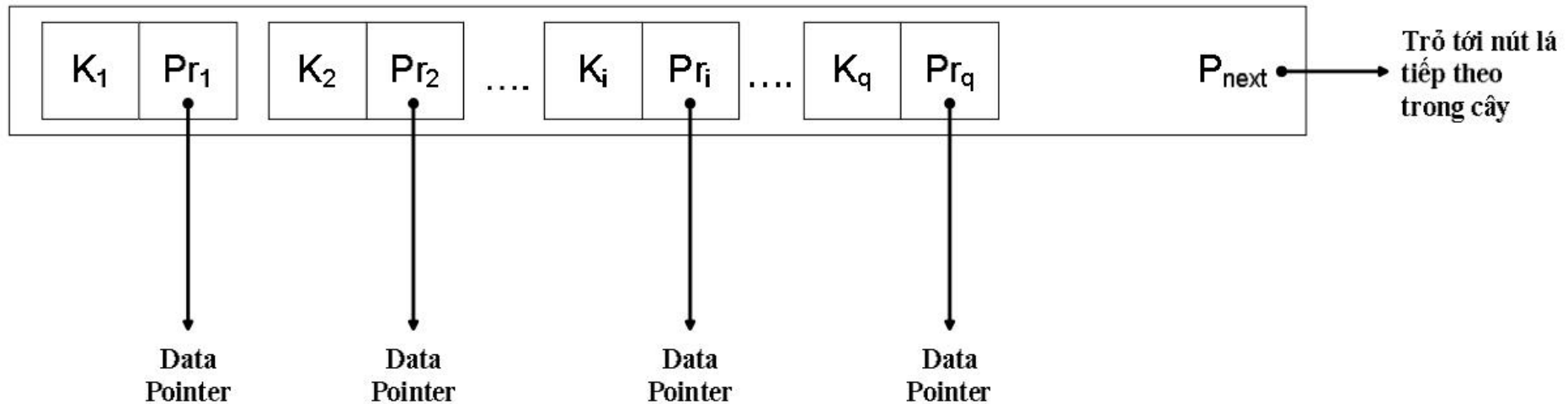
Nút bên trái sẽ có tất cả các giá trị **nhỏ hơn** hoặc **bằng nút gốc**



6.1.2 File chỉ mục B⁺ -Cây (tt)

❖ Cấu trúc của một nút lá B⁺ Tree có bậc p

- Con trỏ dữ liệu chỉ xuất phát ở nút lá
- Các Nút lá liên kết với nhau theo thứ tự khóa tìm kiếm





6.1.2 File chỉ mục B⁺ -Cây (tt)

- Là một dạng **cây cân bằng** (mọi đường dẫn từ gốc đến lá có cùng độ dài)
- Mỗi **nút không là lá** có **số con (pointer)** nằm trong khoảng **$[m/2]..m$** , m là 1 số cố định gọi là bậc của cây
- Mỗi **nút lá** có **số giá trị (key)** nằm trong khoảng **$[(m-1)/2] .. m-1$**
- Nút gốc (nếu không phải nút lá) có ít nhất 2 nút con
- Mọi nút lá đều nằm cùng một mức



File chỉ mục B⁺ -cây - THÊM

- **Ý tưởng:** Tìm vị trí khóa có thể thêm vào cây. Khóa mới sẽ được thêm vào nút lá:
 - Nếu chưa đầy → Thêm khóa mới vào là hoàn tất
 - Nếu đầy → Tách nút lá ra làm hai nút cạnh nhau trong cùng một mức. Cụ thể: chuyển phần tử giữa lên nút cha
- *Quá trình tách nút có thể được lan truyền ngược về nút gốc và kết thúc khi có một nút cha nào đó cần được thêm 1 khóa từ dưới lên mà chưa đầy*



Chèn (xen) khóa vào B⁺ -cây

Ví dụ:

- Cho B⁺ -cây bậc 3 rỗng
- Hãy xây dựng B⁺ Tree bằng việc thêm lần lượt các khóa theo thứ tự sau:

27	19	23	9	1	3	11	21	5	13	17	15	29	25
----	----	----	---	---	---	----	----	---	----	----	----	----	----

	Công thức		m = 3	
Nút	Ít nhất	Nhiều nhất	Ít nhất	Nhiều nhất
Nút gốc	1		1	
Nút trong	m/2	m	3/2= 1.5 → 2	3
Nút lá	(m-1)/2	m-1	(3-1)/2= 1	2

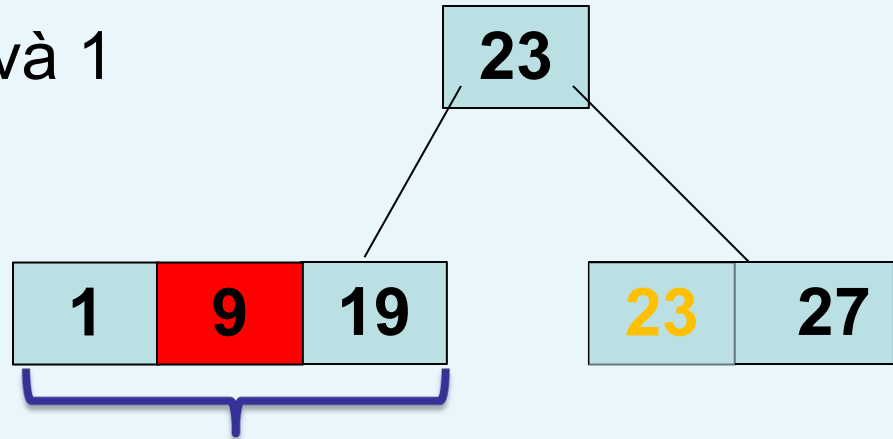
27	19	23	9	1	3	11	21	5	13	17	15	29	25
----	----	----	---	---	---	----	----	---	----	----	----	----	----



Tách nút



Thêm nút: 9 và 1

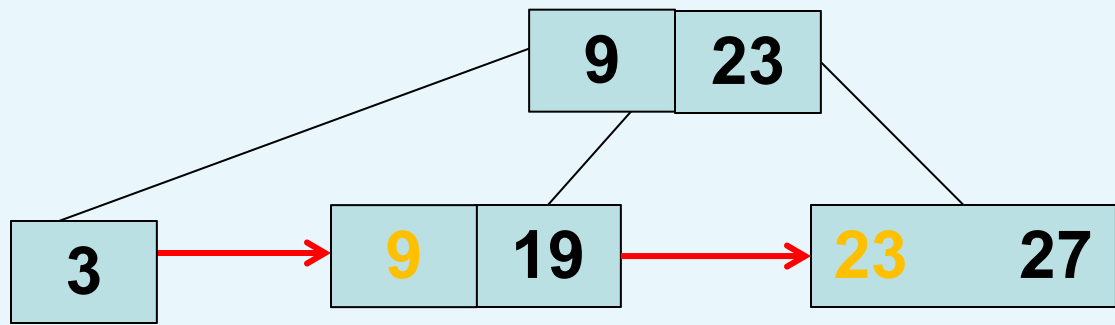


Tách nút

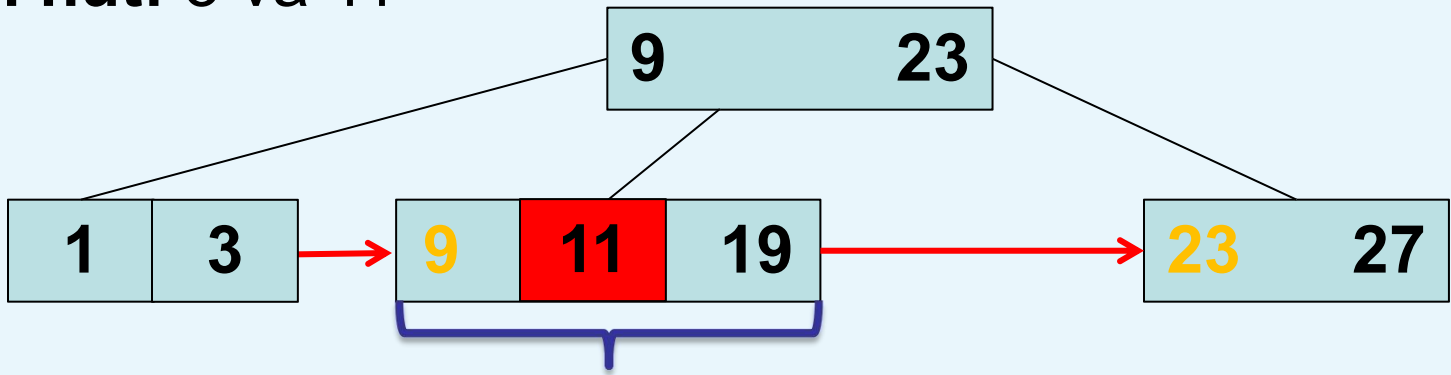


27	19	23	9	1	3	11	21	5	13	17	15	29	25
----	----	----	---	---	---	----	----	---	----	----	----	----	----

CANTHO UNIVERSITY



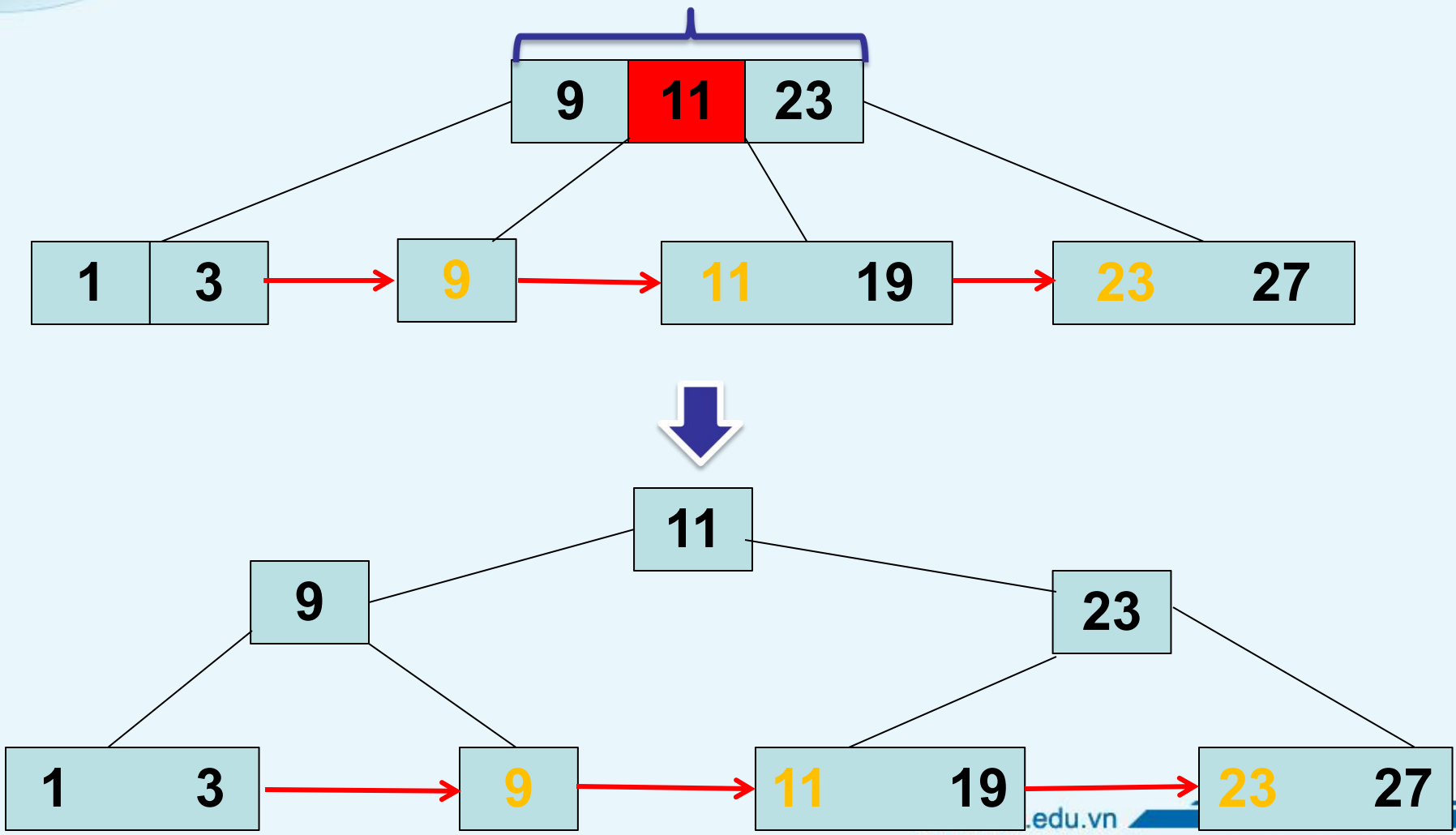
Thêm nút: 3 và 11



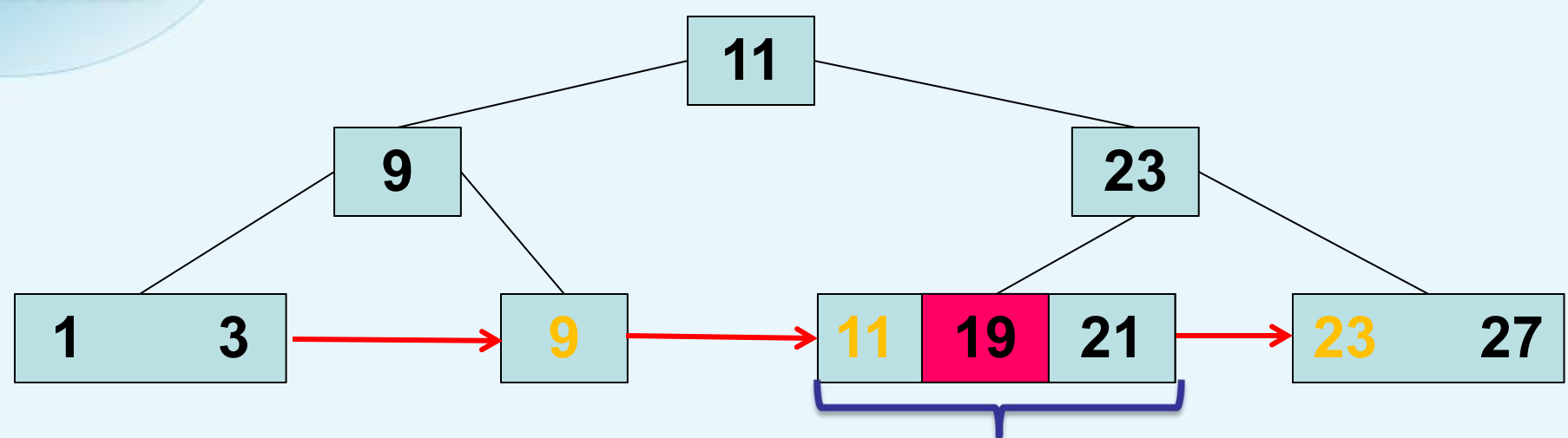
Tách nút

27	19	23	9	1	3	11	21	5	13	17	15	29	25
----	----	----	---	---	---	----	----	---	----	----	----	----	----

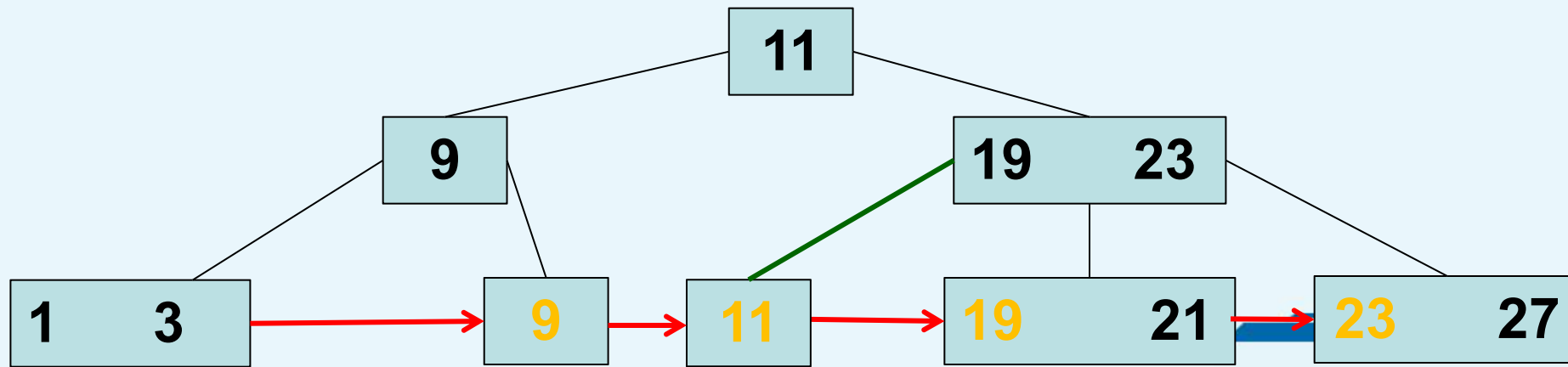
Tách nút



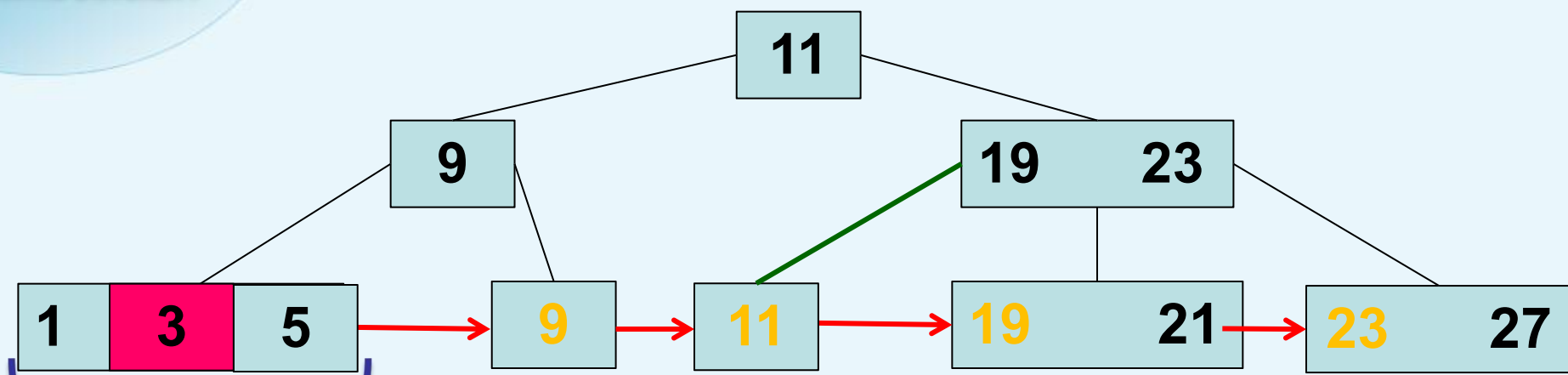
27	19	23	9	1	3	11	21	5	13	17	15	29	25
----	----	----	---	---	---	----	----	---	----	----	----	----	----



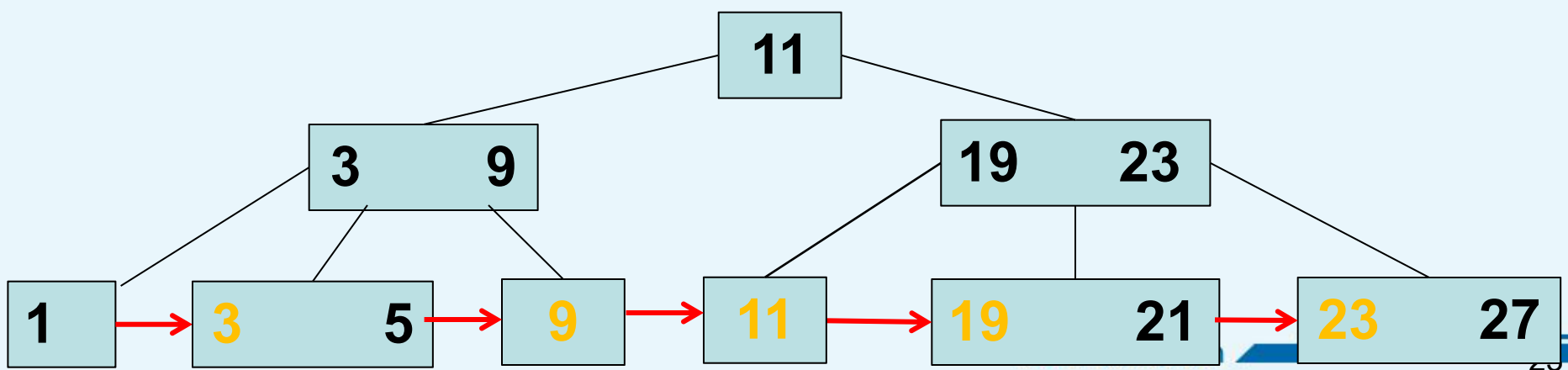
Tách nút



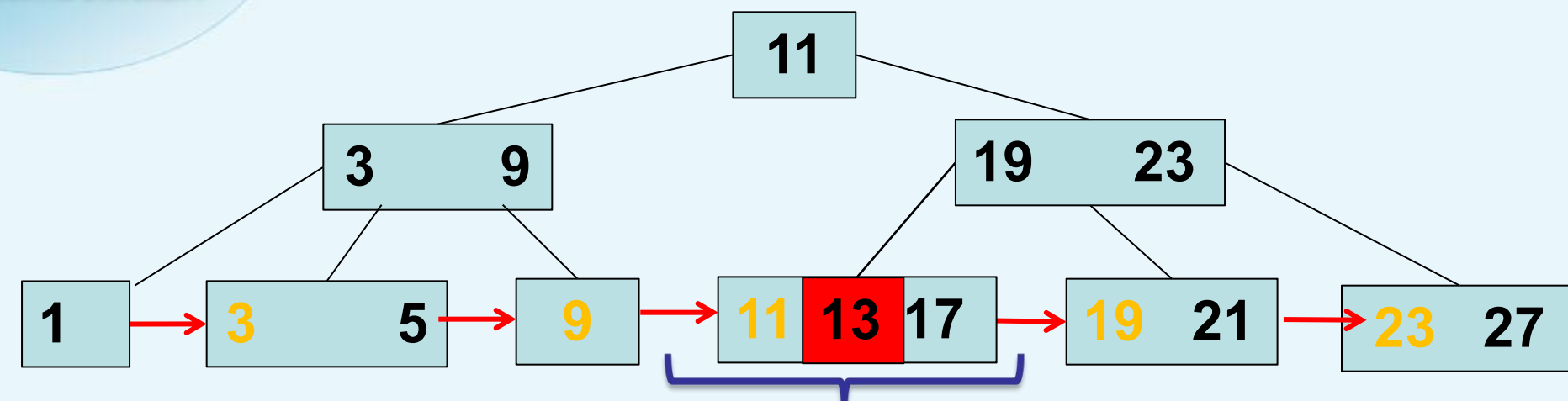
27	19	23	9	1	3	11	21	5	13	17	15	29	25
----	----	----	---	---	---	----	----	---	----	----	----	----	----



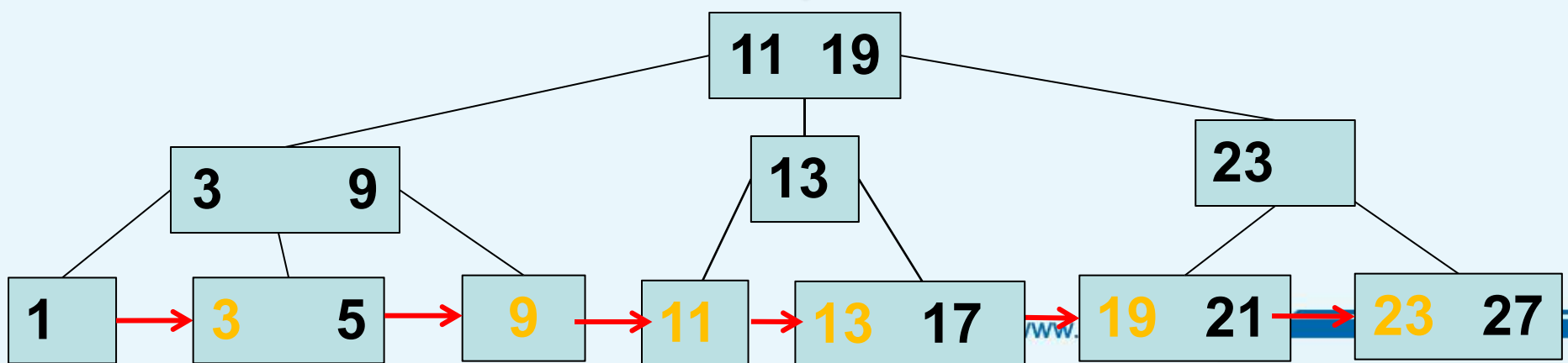
Tách nút



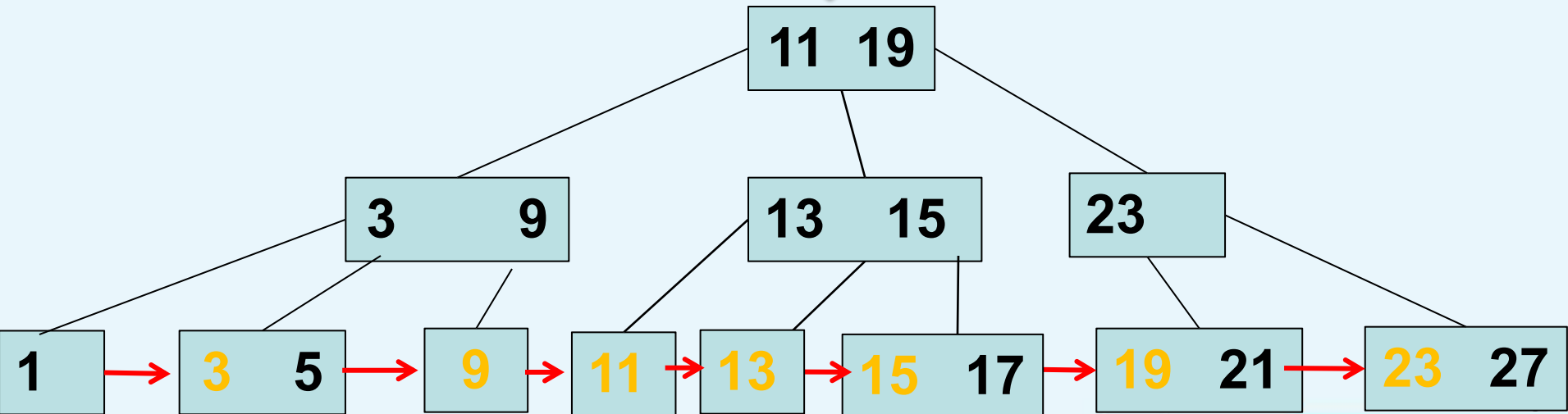
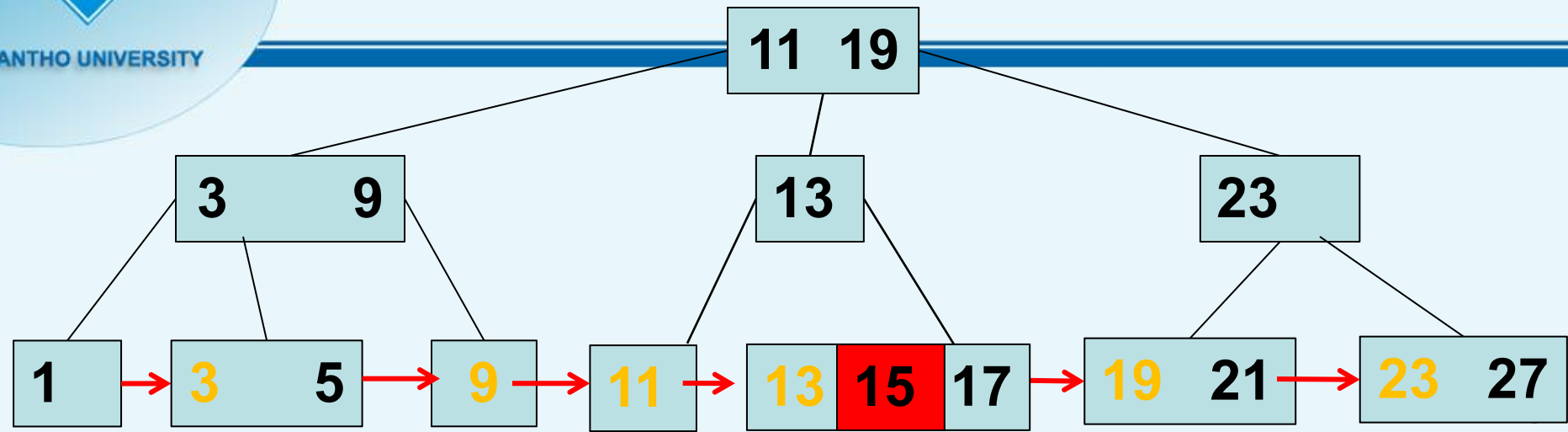
27	19	23	9	1	3	11	21	5	13	17	15	29	25
----	----	----	---	---	---	----	----	---	----	----	----	----	----



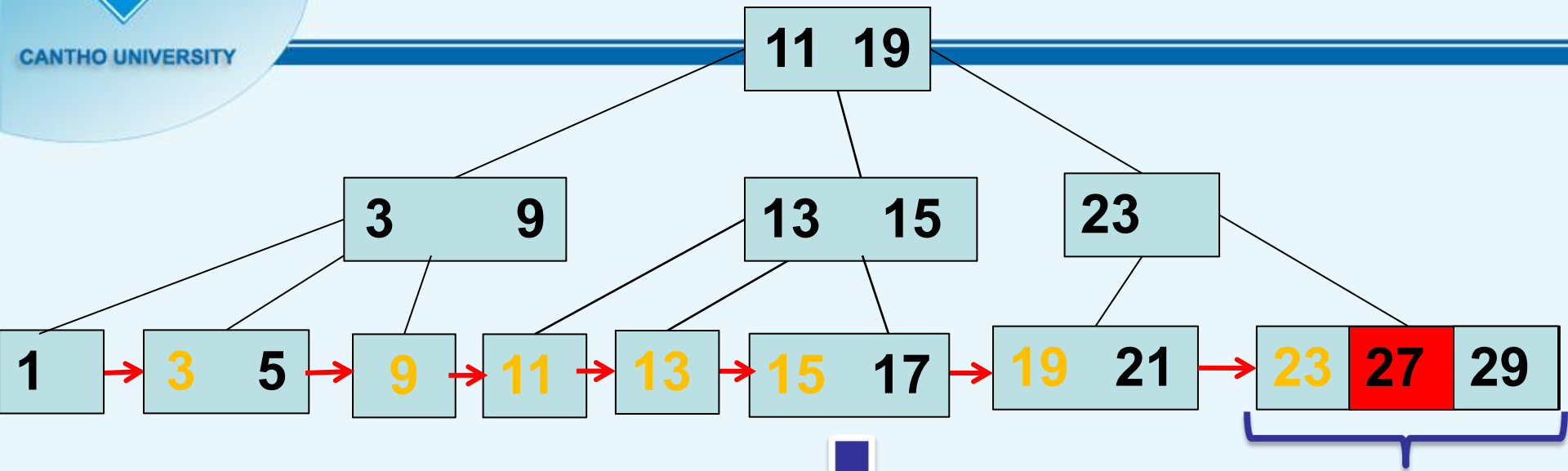
Tách nút



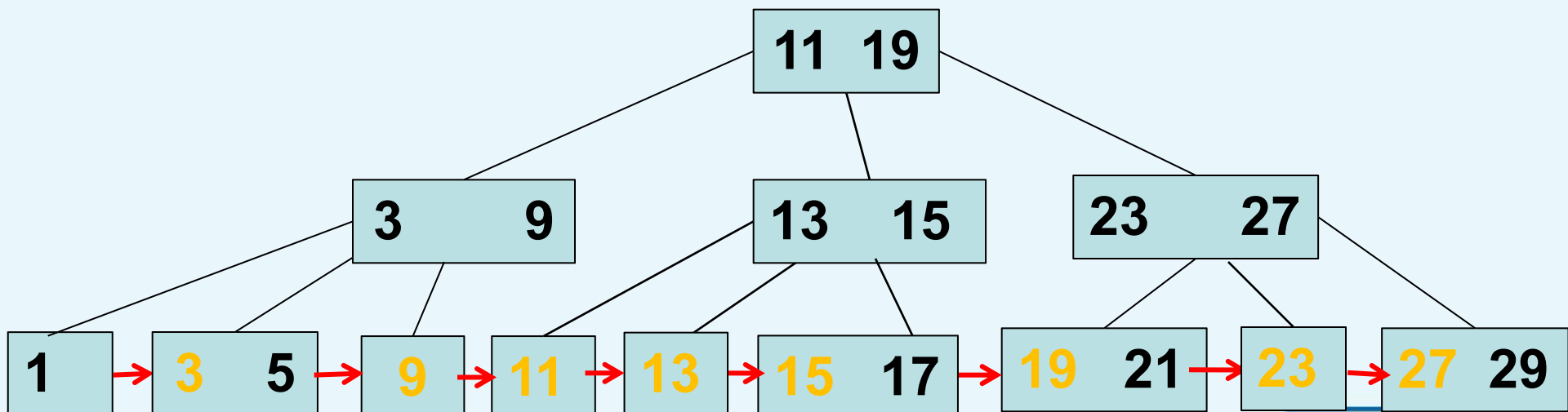
27	19	23	9	1	3	11	21	5	13	17	15	29	25
----	----	----	---	---	---	----	----	---	----	----	----	----	----



27	19	23	9	1	3	11	21	5	13	17	15	29	25
----	----	----	---	---	---	----	----	---	----	----	----	----	----



Tách nút



27	19	23	9	1	3	11	21	5	13	17	15	29	25
----	----	----	---	---	---	----	----	---	----	----	----	----	----

Kết quả cuối cùng:

