CANTHO UNIVERSITY

CHƯƠNG 4

CẤU TRÚC DỮ LIỆU VÀ THUẬT TOÁN LƯU TRỮ NGOÀI

Bộ môn CÔNG NGHỆ PHẦN MỀM
Khoa Công nghệ thông tin và Truyền thông
Đại học Cần Thơ

Võ Huỳnh Trâm

NỘI DUNG

- Mô hình và đánh giá các xử lý ngoài.
- *Sắp xếp ngoài.*
- Lưu trữ thông tin trong tập tin:
  - Tập tin tuần tự
  - Tập tin bảng băm
  - Tập tin chỉ mục
  - **Tập tin B-cây**

Tại sao phải xử lý ngoài ?

- Trong các thuật toán đề cập trước đây, ta đã giả sử rằng số lượng dữ liệu đầu vào khá nhỏ có thể chứa hết ở *bộ nhớ trong* (main memory).
- **Vấn đề**: Đối với bài toán có số lượng dữ liệu vượt quá khả năng lưu trữ của bộ nhớ trong. Chẳng hạn: xử lý *phiếu điều tra dân số toàn quốc* hay *thông tin về quản lý đất đai cả nước* ? $\Rightarrow$ Dùng ***bộ nhớ ngoài*** để lưu trữ và xử lý.
- Các thiết bị lưu trữ ngoài như *băng từ, đĩa từ* đều có khả năng lưu trữ lớn nhưng đặc điểm truy nhập hoàn toàn khác với bộ nhớ trong
  $\rightarrow$ Cần tìm **cấu trúc dữ liệu** và **thuật toán** thích hợp xử lý dữ liệu lưu trữ trên *bộ nhớ ngoài* ?

BỘ NHỚ NGOÀI

Đĩa mềm

SDD

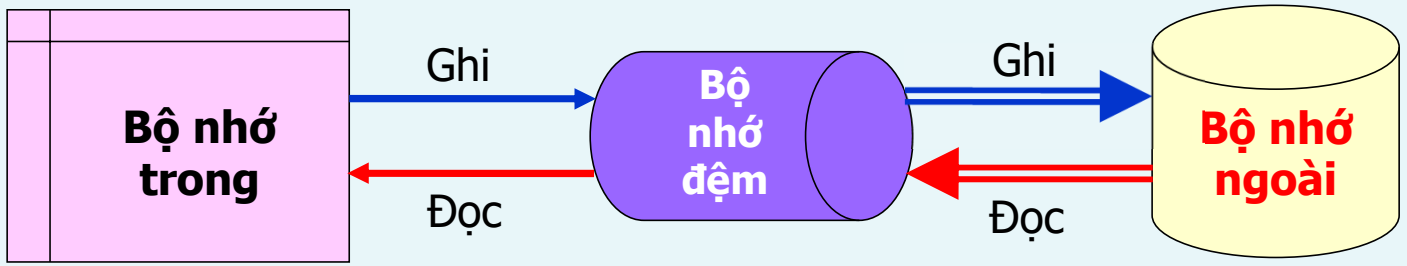
Ổ cứng

USB

CD

Mô hình xử lý ngoài

- Hệ điều hành chia *bộ nhớ ngoài* thành các **khối (block)** có kích thước bằng nhau, kích thước này thay đổi tùy thuộc vào hệ điều hành (khoảng từ 512 bytes đến 4096 bytes.)
- Có thể xem một ***tập tin*** bao gồm nhiều ***mẩu tin*** được lưu trong các **khối.**
- Mỗi khối lưu một số nguyên vẹn các *mẩu tin*.
- **Kiểu dữ liệu *tập tin*** thích hợp nhất cho việc biểu diễn dữ liệu lưu trong bộ nhớ ngoài.



Mỗi lần truy xuất 1 mẩu tin **Mỗi lần truy xuất 1 khối**

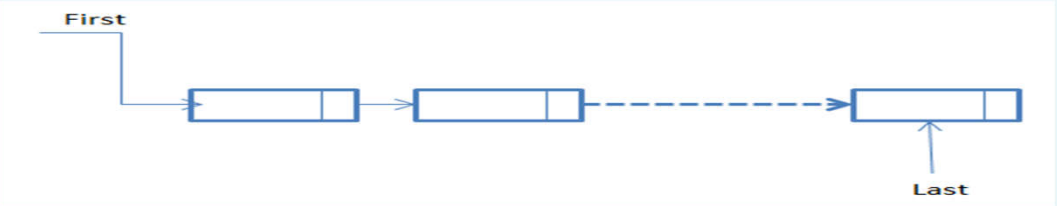
Đánh giá các thuật toán xử lý ngoài

- Đối với bộ nhớ ngoài, thời gian tìm đọc khối vào bộ nhớ trong là **rất lớn** so với thời gian thao tác trên dữ liệu trong khối đó → Chúng ta tập trung vào việc xét ***số lần đọc khối*** vào bộ nhớ trong và ***số lần ghi khối*** ra bộ nhớ ngoài, hay phép **truy xuất khối** (block access).
- Nếu số lần truy xuất khối ít thì thuật toán có hiệu quả.
- Để cải tiến thuật toán, không thể tìm cách tăng kích thước khối (vì kích thước các khối là cố định) mà phải tìm cách **giảm số lần truy xuất khối.**

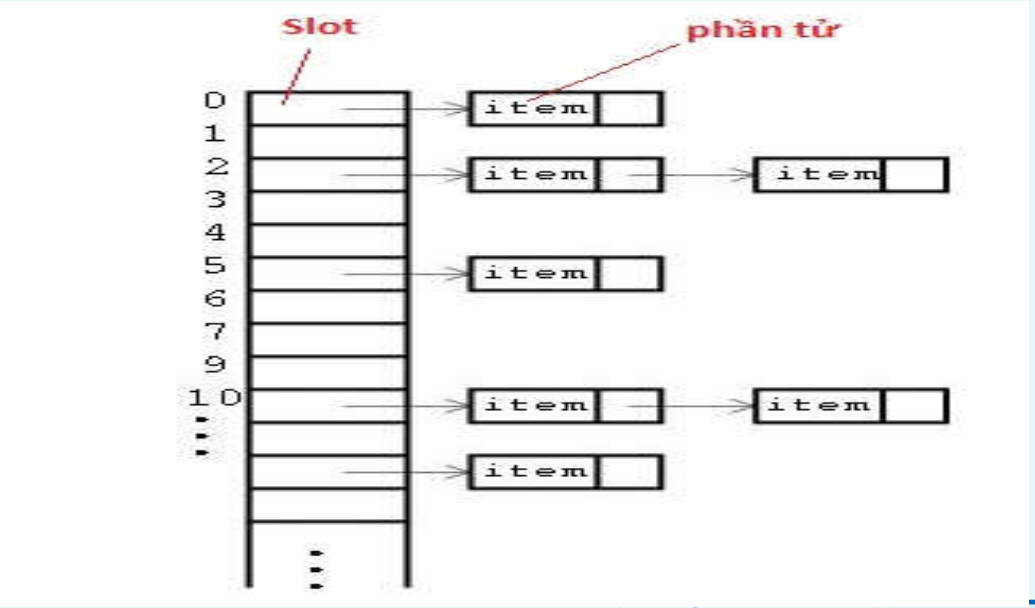
CANTHO UNIVERSITY

CÁC HÌNH THỨC TỔ CHỨC TẬP TIN

– Tập tin **tuần tự**
(Sequential File)



– Tập tin **bảng băm**
(Hash File)

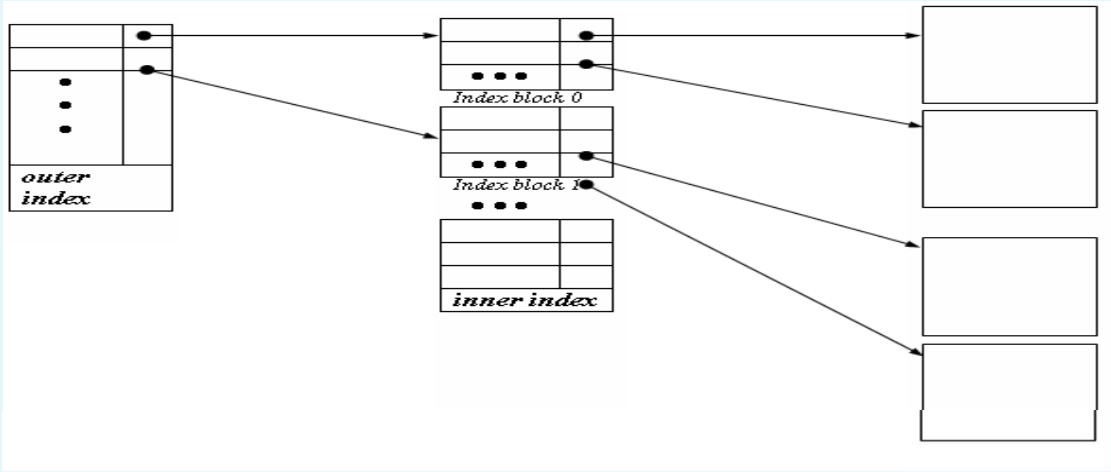


CANTHO UNIVERSITY

CÁC HÌNH THỨC TỔ CHỨC TẬP TIN

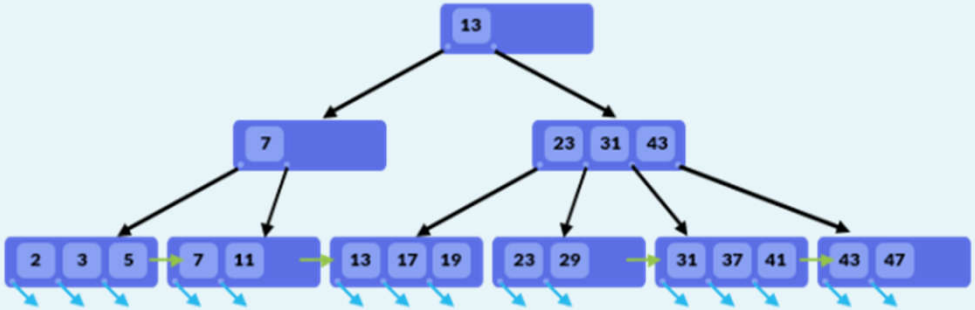
– Tập tin **chỉ mục**

(Index File)



– Tập tin **B-cây**

(B-tree)



Cây tìm kiếm m-phân (M-ary tree): Tổ chức

- **Cây tìm kiếm m-phân (m-ary tree) / Cây tìm kiếm đa phân** là sự tổng quát hoá của *cây tìm kiếm nhị phân* trong đó mỗi nút có thể có m nút con.
- Giả sử n_1 và n_2 là hai con của một nút nào đó, n_1 **bên trái** n_2 thì tất cả các *con của* n_1 *có giá trị* < *giá trị của các nút con của* n_2.

B – cây (B – trees): ĐỊNH NGHĨA

- **B-cây bậc m** là ***Cây tìm kiếm m-phân cân bằng*** có các tính chất sau:

  (1) Nút gốc hoặc là *lá* hoặc có *ít nhất* ***2 nút con***

  (2) Mỗi **nút**, trừ nút gốc và nút lá, có **từ ⌈m/2⌉ đến m** nút con

  (3) Các đường đi từ gốc tới lá có **cùng độ dài**

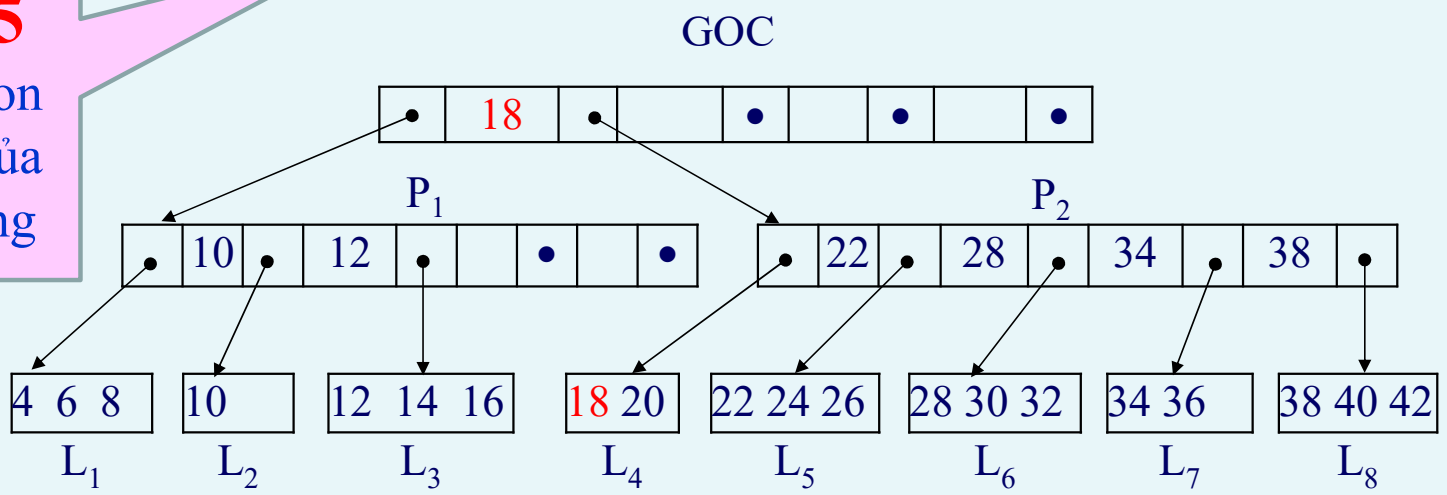
  (4) Các khóa và cây con sắp xếp theo **cây tìm kiếm**

Tập tin B – cây: Ví dụ

Ví dụ: Tập tin 20 mẩu tin với giá trị khóa là số nguyên được tổ chức thành **B-cây bậc 5**, nút lá chứa được nhiều nhất **3 mẩu tin**.

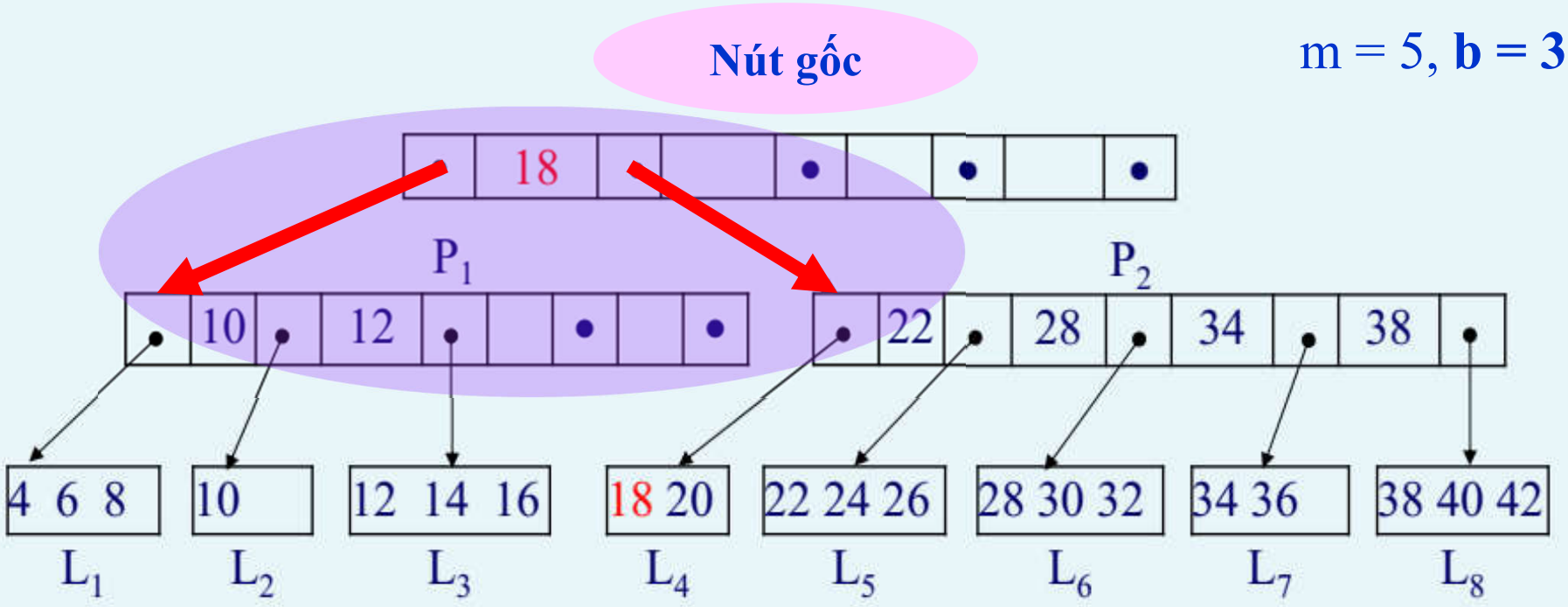
m = 5
m: số con tối đa của nút trong

b = 3
b: số mẩu tin tối đa trong nút lá



Tập tin B – cây: Tính chất

(1) Nút gốc hoặc là *lá* hoặc có *ít nhất* ***2 nút con***

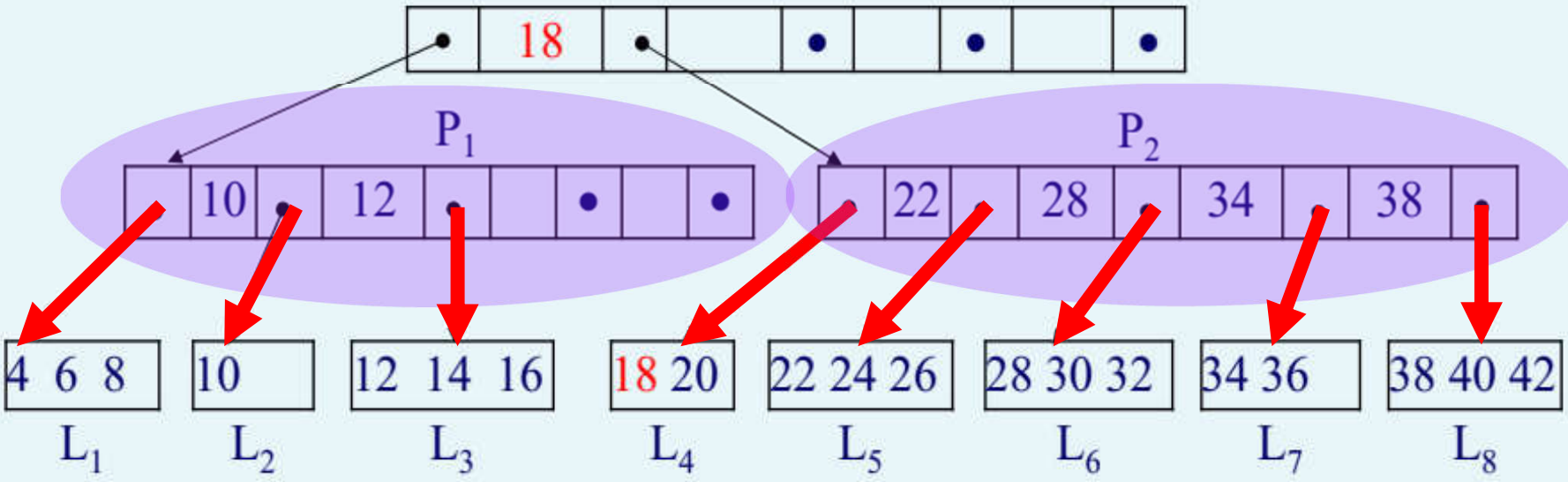


Tập tin B – cây: Tính chất

(2) Mỗi **nút**, trừ nút gốc và nút lá, có $\lceil m/2 \rceil \rightarrow m$ nút con

Nút trong

GOC $m = 5 \Rightarrow \lceil m/2 \rceil = \lceil 5/2 \rceil = 3 \rightarrow 5$ con

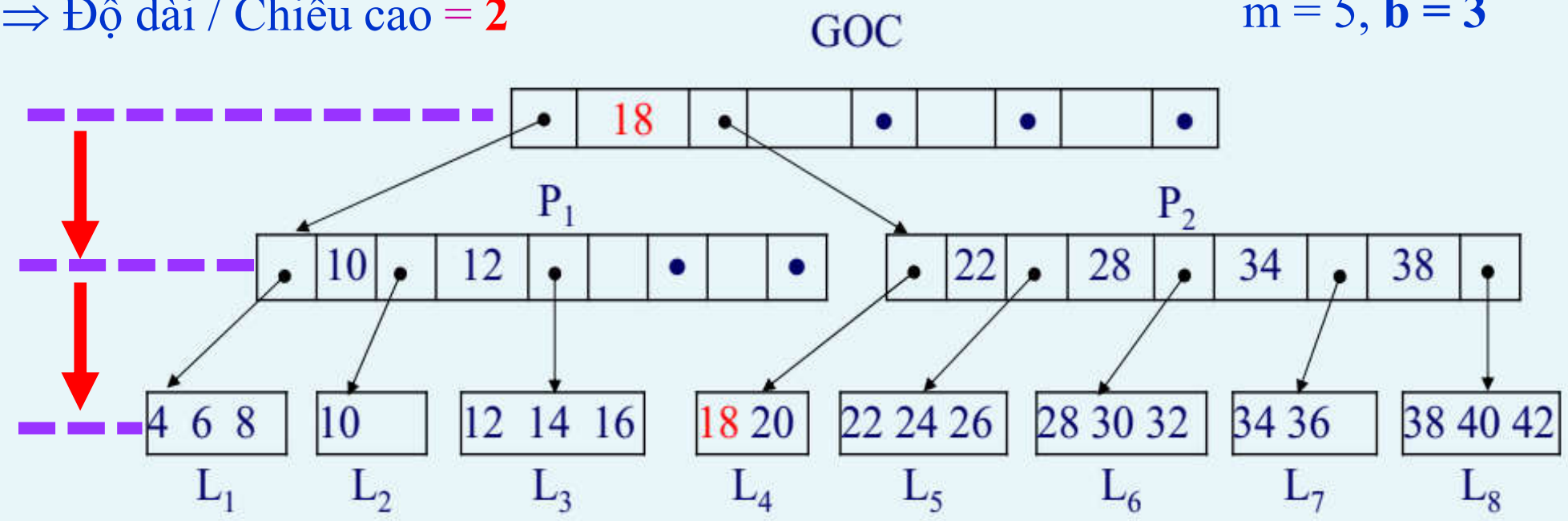


Tập tin B – cây: Tính chất

(3) Các đường đi từ gốc tới lá có **cùng độ dài.**

⇒ Độ dài / Chiều cao = **2**

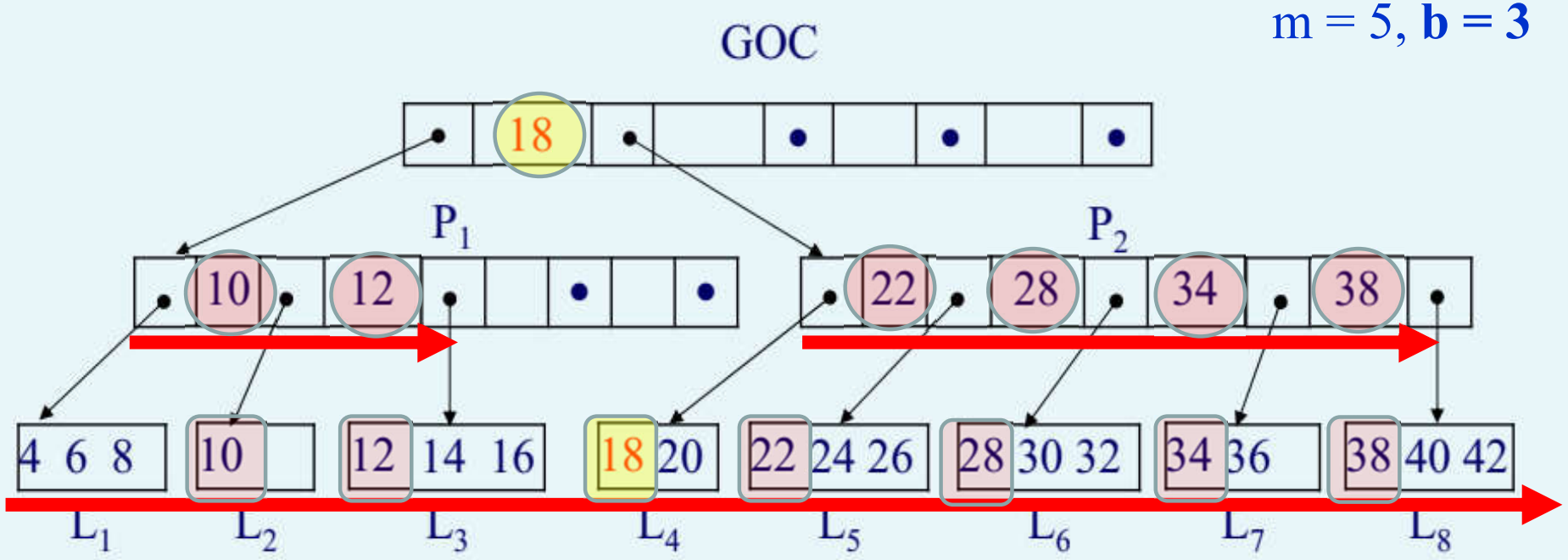
m = 5, **b = 3**



Tập tin B – cây: Tính chất

(4) Các khóa và cây con sắp xếp theo **cây tìm kiếm**

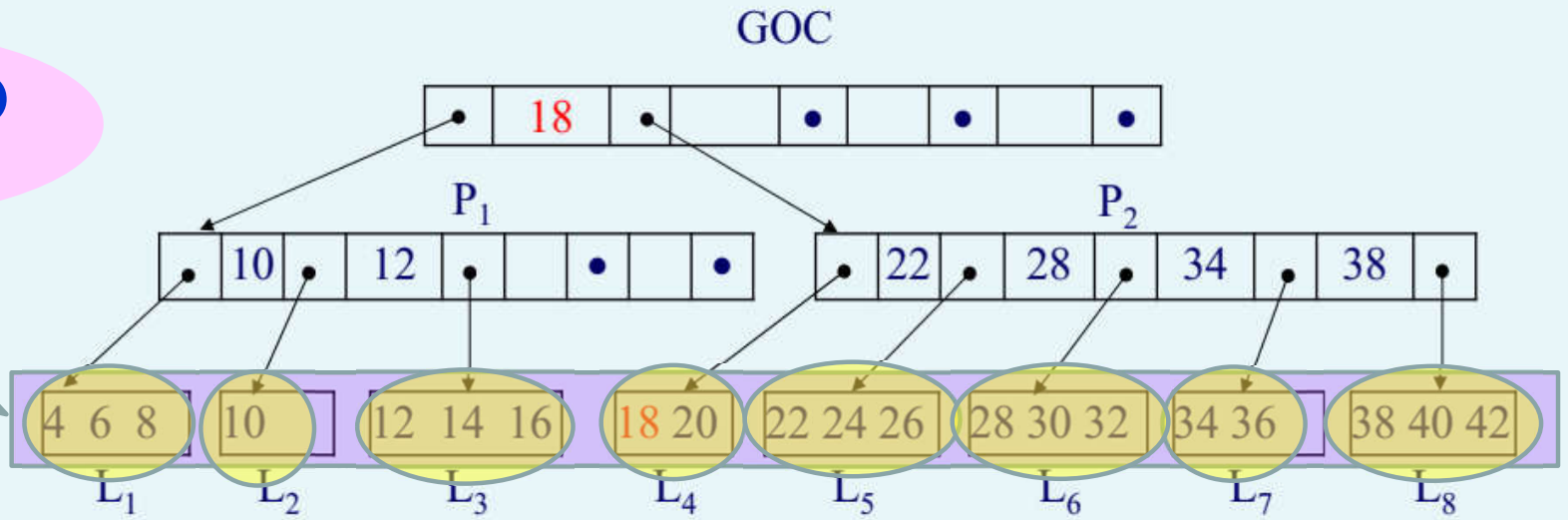
m = 5, **b = 3**



Tập tin B – cây: Tổ chức

- Mỗi **nút** trên cây là một **khối** trên đĩa, các mẩu tin của tập tin được lưu trữ trong các nút lá trên B-cây theo thứ tự của khoá.
- Nút lá lưu trữ được nhiều nhất **b mẩu tin.** m = 5, **b = 3**

1 khối (block) trên đĩa



Tập tin F có n = 20 mẩu tin chứa trong 8 lá

Tập tin B – cây: Tổ chức

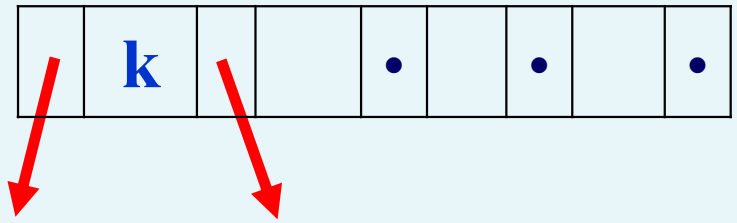
Giả sử **B-cây bậc 5** với các nút lá chứa được nhiều nhất 3 **mẩu tin**

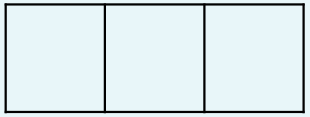
m = 5

b = 3

Nút gốc

Nút lá





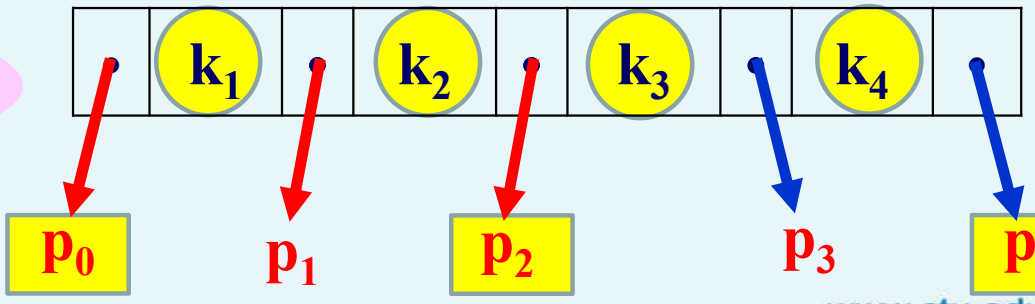
Tập tin B – cây: Tổ chức

- **Mỗi nút** không phải là nút lá có dạng $(p_0, k_1, p_1, k_2, p_2, \dots, k_n, p_n)$, với p_i $(0 \le i \le n)$ là con trỏ, trỏ tới nút con thứ i và k_i là giá trị khóa

  Các khoá trong một nút được sắp thứ tự: $k_1 < k_2 < \dots < k_n$

  - Tất cả các khoá trong cây con được trỏ bởi p_0 đều < k_1
  - Tất cả các khoá trong cây con được trỏ bởi p_i $(0 < i < n)$ đều $\ge k_i$ và $< k_{i+1}$
  - Tất cả các khoá trong cây con được trỏ bởi p_n đều $\ge k_n$

Nút trong



$m = 5, b = 3$

$\Rightarrow \lceil m/2 \rceil \rightarrow$ m con

$\Rightarrow 3 \rightarrow 5$ con

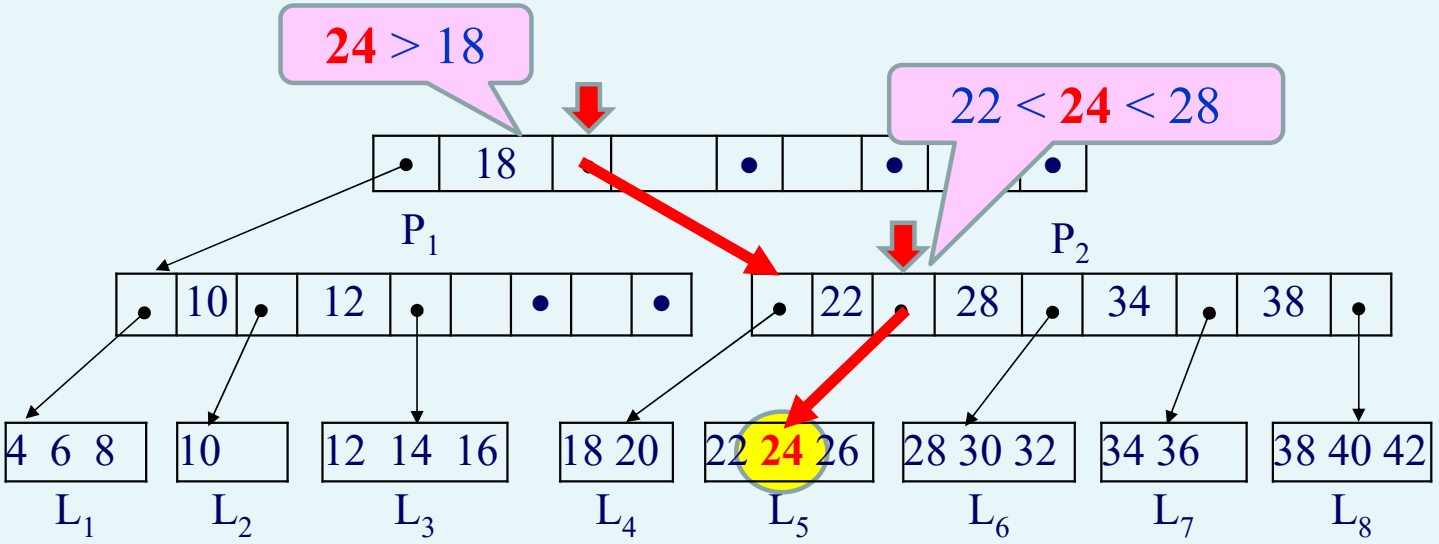
Tập tin B - cây : TÌM MẪU TIN

Tìm mẩu tin: Bắt đầu từ nút gốc đến nút lá chứa r (nếu r tồn tại trong tập tin).

- Tại mỗi bước, đưa nút trong **$(p_0, k_1, p_1, k_2, p_2, ..., k_n, p_n)$** vào bộ nhớ trong và xác định mối quan hệ giữa x với các giá trị khóa k_i.
  - Nếu **$k_i \le x < k_{i+1}$** $(0 < i < n)$: xét tiếp nút được trỏ bởi p_i.
  - Nếu **$x < k_1$** : xét tiếp nút được trỏ bởi p_0.
  - Nếu **$x \ge k_n$** : xét tiếp nút được trỏ bởi p_n.
- Quá trình trên sẽ dẫn đến việc xét nút lá. Tại nút lá này, tìm mẩu tin r với khóa x bằng *tìm kiếm tuần tự* hoặc *tìm kiếm nhị phân*.

Ví dụ tìm mẩu tin

Ví dụ: Tìm mẩu tin r với khóa x = **24** trong tập tin được biểu diễn trong hình sau:



Tập tin B - cây : XEN MẨU TIN

Xen mẩu tin: Tìm r. Việc tìm kiếm ẽ dẫn đến nút lá L.

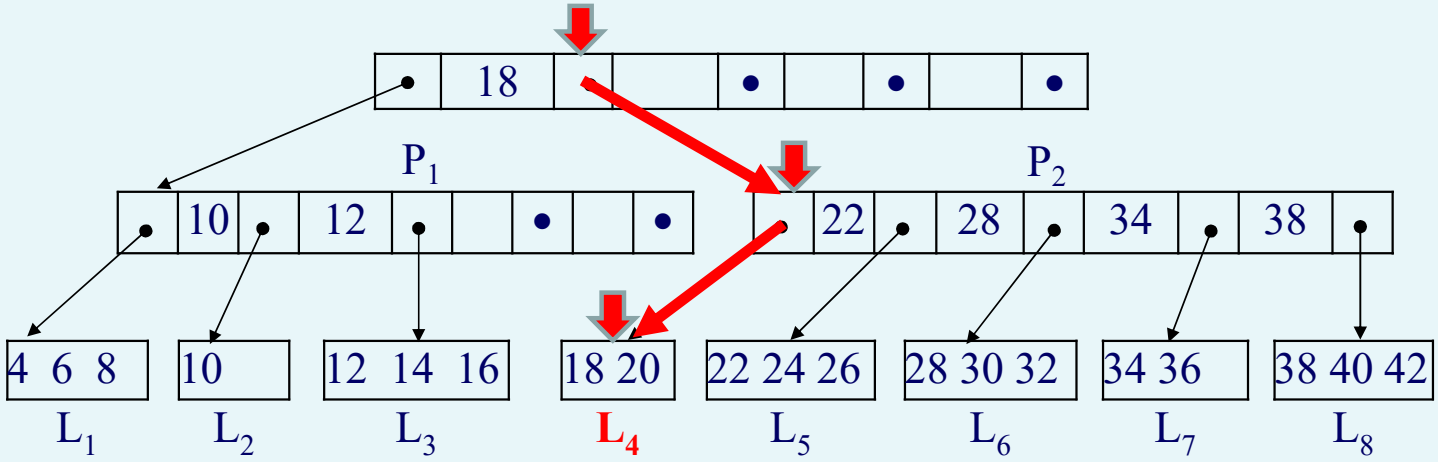
- *Nếu tìm thấy*, thông báo "Mẩu tin đã tồn tại",
- *Ngược lại* thì L là nút lá có thể xen r vào trong đó.

(1) **Nếu L còn chỗ**: thêm r vào đúng thứ tự và kết thúc.

(2) **Nếu L không còn chỗ**: cấp phát khối mới L', dời $\lceil b/2 \rceil$ mẩu tin cuối L sang L', *xen r vào L hoặc L' sao cho đảm bảo thứ tự các khoá trong khối*.

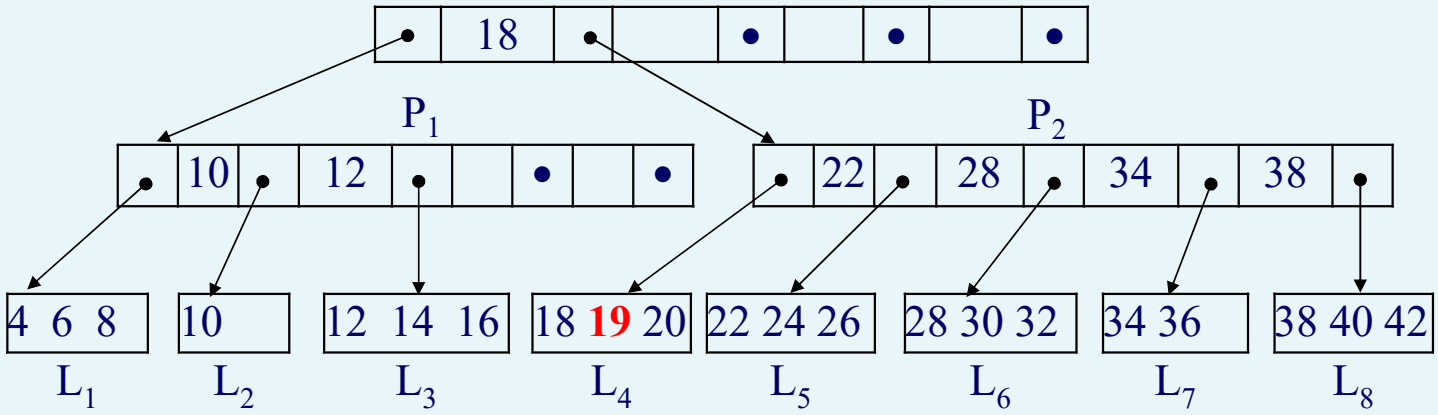
Ví dụ xen mẩu tin mới (1)

Ví dụ : Thêm mẩu tin r với khóa x = **19** vào tập tin được biểu diễn trong hình sau:



Ví dụ xen mẩu tin mới (1)

Kết quả: Mẩu tin r với khóa x = **19** đã được thêm vào :



Tập tin B - cây : XEN MẨU TIN

Xen mẩu tin: (2) **Nếu L không còn chỗ**:

- Cấp phát khối lá mới L'
- Chuyển $\lceil b/2 \rceil$ mẩu tin cuối của L sang L'
- *Xen r vào L hoặc L' sao cho đảm bảo thứ tự các khoá trong khối*
- Xen **đệ quy** cặp khóa - con trỏ của L' vào nút cha P của nó

Trường hợp P đã có đủ m con:

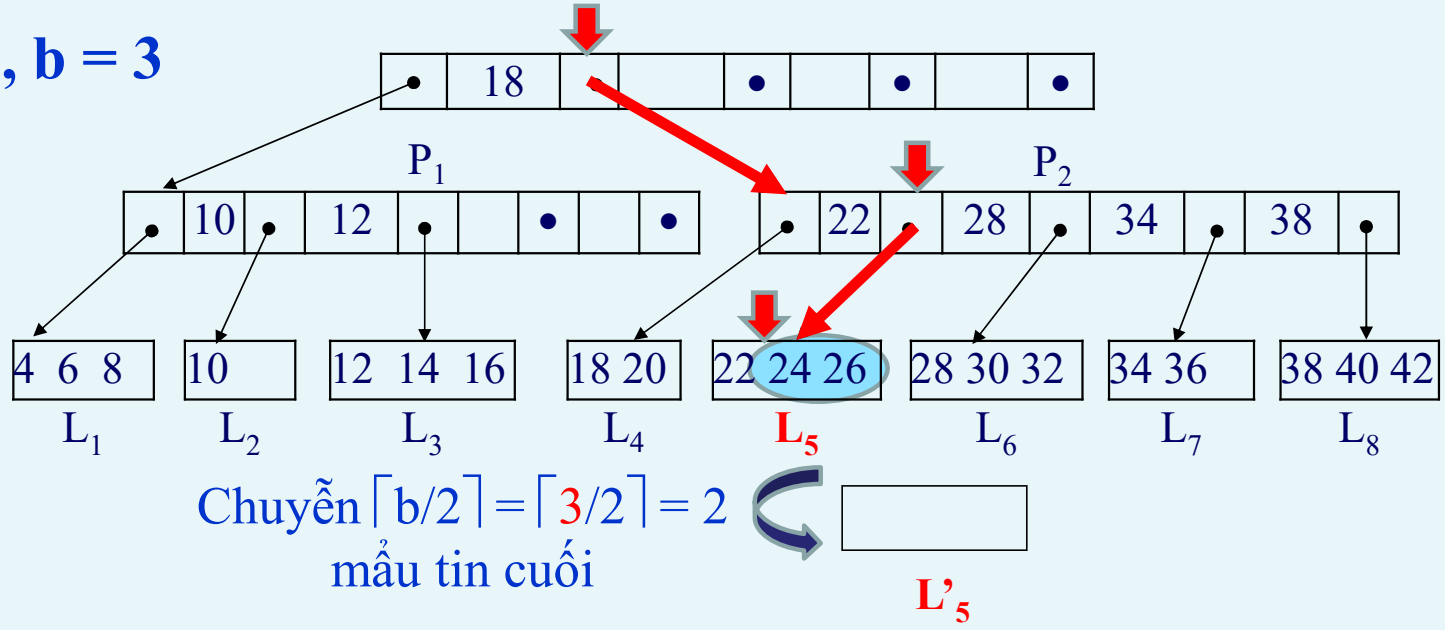
- Cấp phát thêm khối mới P'
- Chuyển $\lceil m/2 \rceil$ con cuối của P sang P'
- Xen L' vào P hoặc P'
- Xen **đệ quy** cặp khóa - con trỏ của P' vào nút cha của nó…

Quá trình này có thể dẫn tới nút gốc và chia cắt nút gốc, tạo nút gốc mới mà 2 con của nó là 2 nửa nút gốc cũ. Khi đó chiều cao của B-cây sẽ **tăng lên 1**

Ví dụ xen mẩu tin mới (2)

Ví dụ : Xen mẩu tin r với khóa x = **23** vào tập tin được biểu diễn trong hình sau:

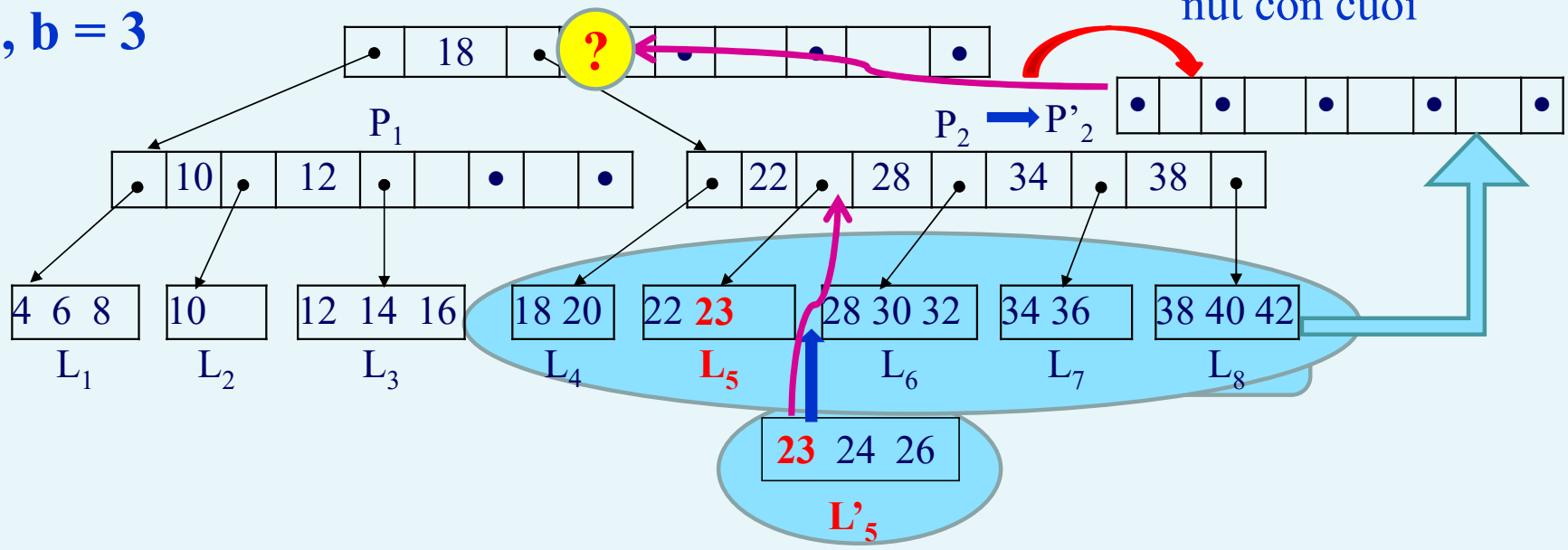
m = 5, b = 3



Ví dụ xen mẩu tin mới (2)

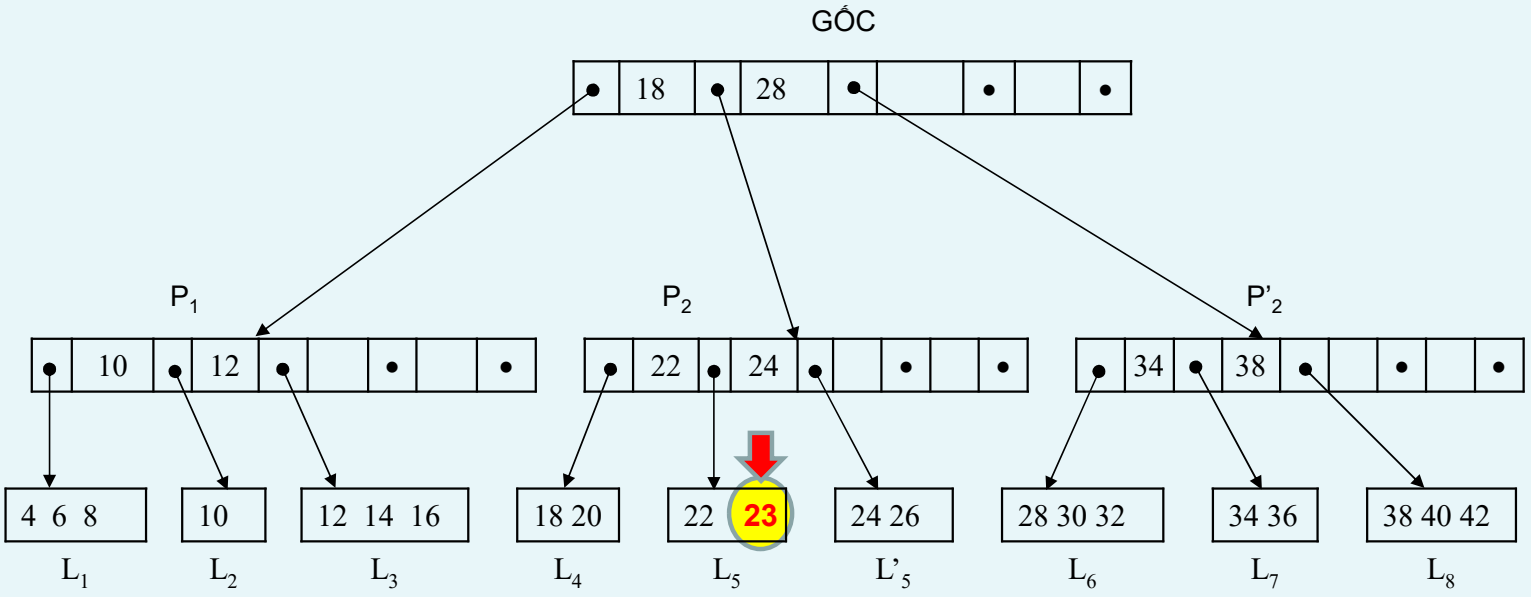
Ví dụ : Xen mẩu tin r với khóa x = **23** vào tập tin được biểu diễn trong hình sau:

m = 5, b = 3



Ví dụ xen mẩu tin mới (2)

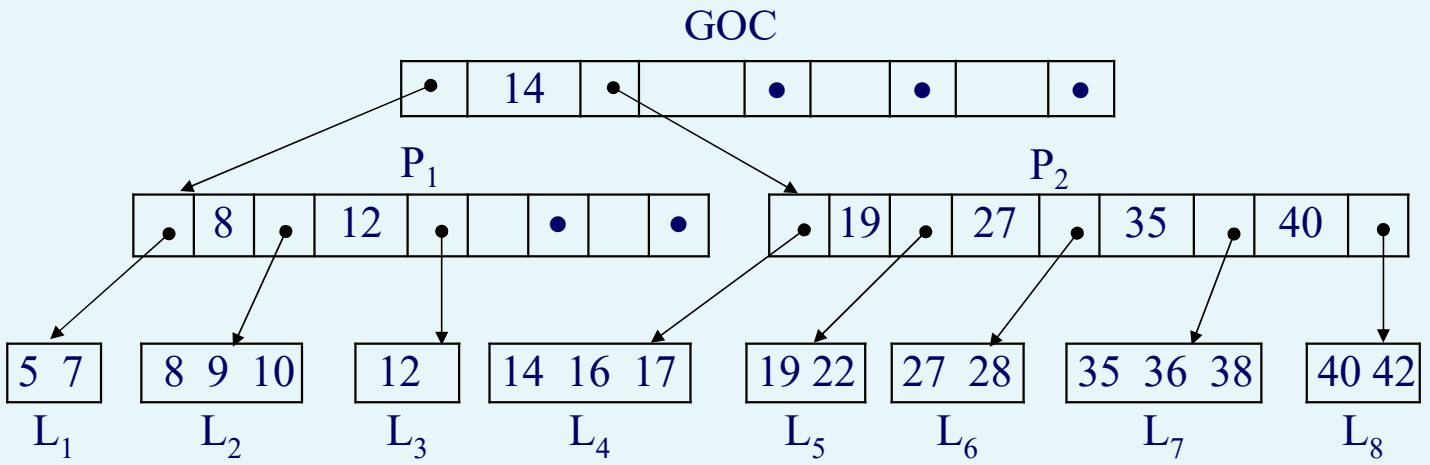
Kết quả : Mẩu tin r với khóa x = **23** đã được thêm vào:



Tập tin B – cây: Bài tập

Bài tập: Cho tập tin bao gồm các mẩu tin với giá trị khóa là các số nguyên được tổ chức thành **B-cây bậc 5** với các nút lá chứa được nhiều nhất **3 mẩu tin** như sau:

1. a) : Thêm mẩu tin r với khóa x = **37** :



Tập tin B - cây : XÓA MẨU TIN

Xóa mẩu tin: Tìm r. Việc tìm kiếm này sẽ dẫn đến nút lá L.

- *Nếu không tìm thấy*, thông báo "Mẩu tin không tồn tại",
- *Ngược lại* thì L là nút lá có thể xóa r trong đó.

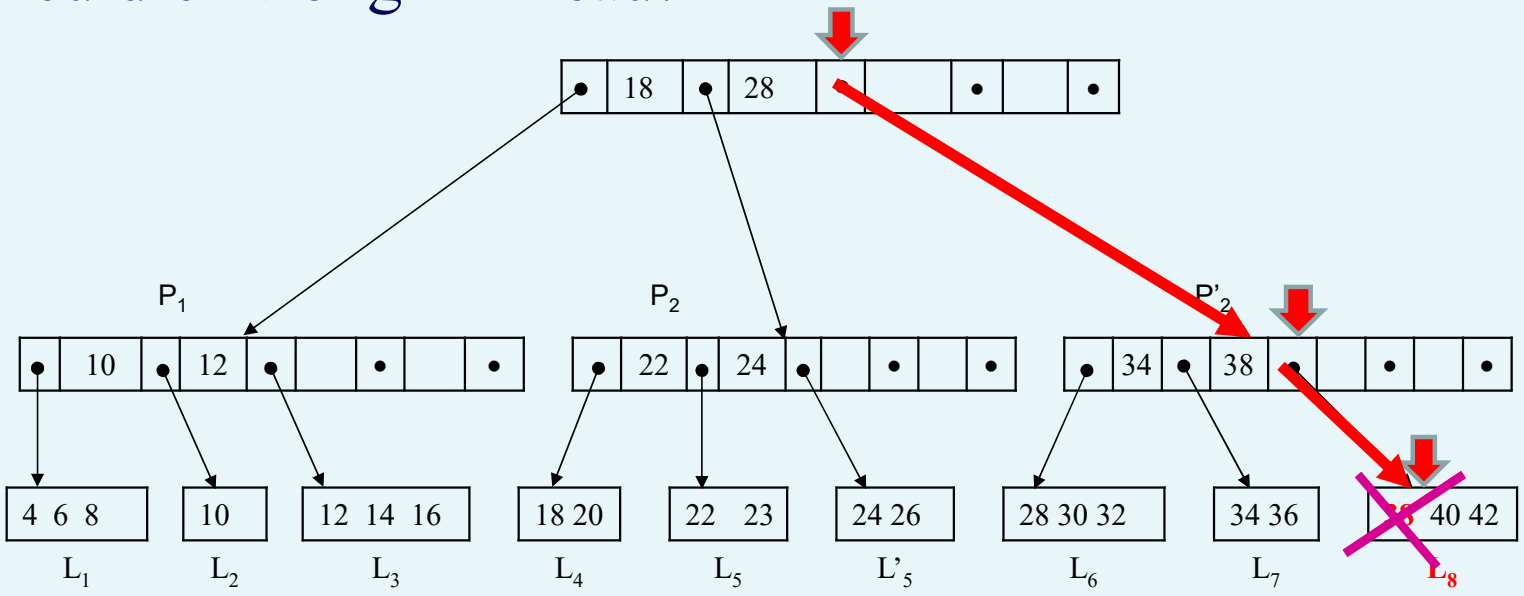
Có 3 trường hợp cần xét :

(1) Nếu r là mẩu tin đầu tiên của L: sau khi xóa, ***đặt lại giá trị khóa*** của L trong P (là giá trị khóa của mẩu tin mới đầu tiên của L).

- L là ***lá trong***: đặt lại giá trị khóa của L trong nút cha P của nó
- L là ***lá bìa*** (*lá con đầu tiên của P):* đặt lại giá trị khóa của L trong *tổ tiên của P.*

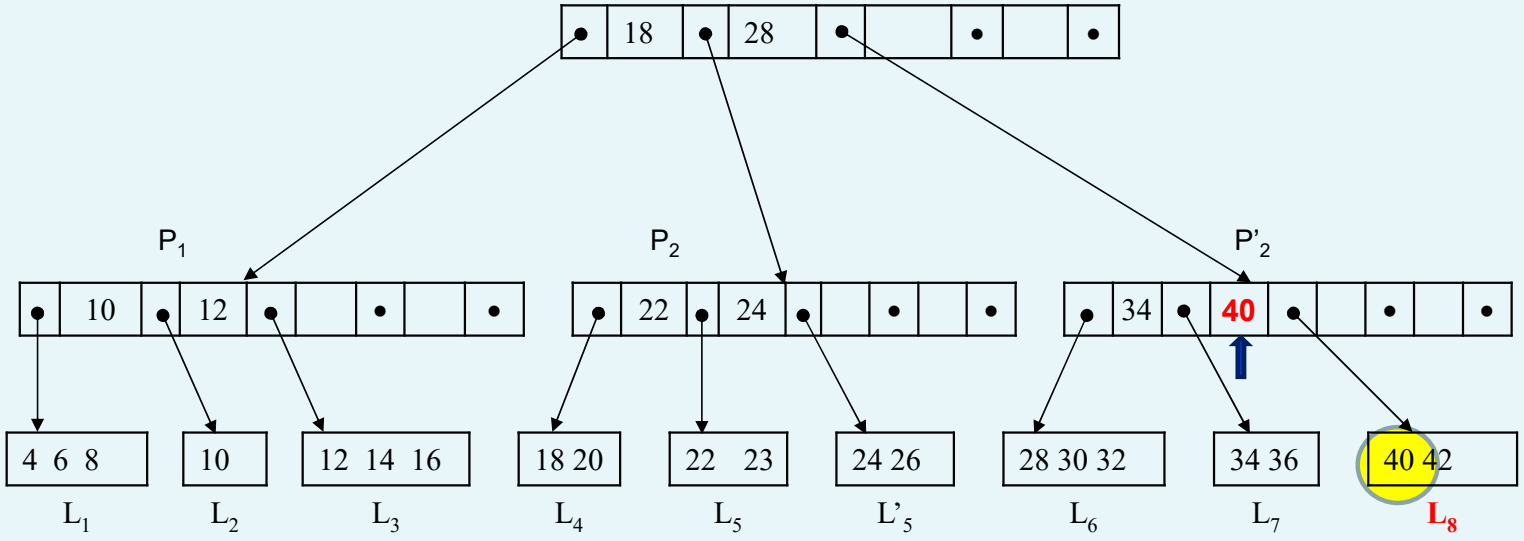
Ví dụ xóa mẩu tin (1)

Ví dụ 1: Xóa mẩu tin r với khóa x = **38** trong tập tin được biểu diễn trong hình sau:



Ví dụ xóa mẩu tin (1)

Kết quả : Mẩu tin r với khóa x = **38** đã được xóa ra khỏi L_8 :



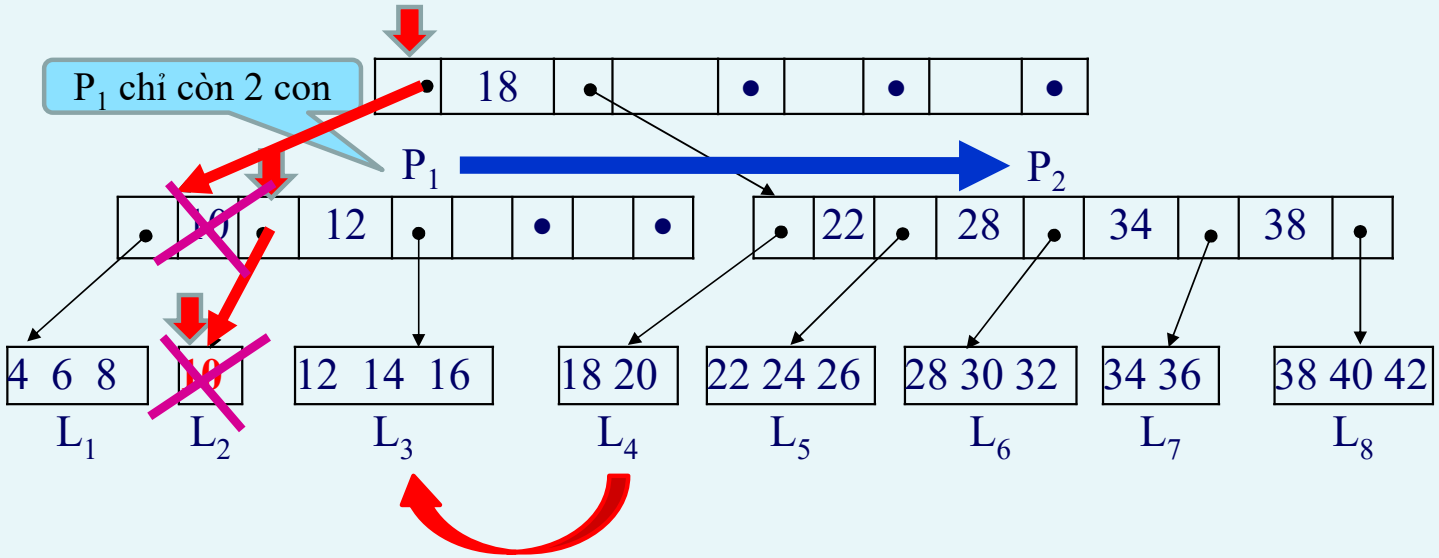
Tập tin B - cây : XÓA MẨU TIN

Xóa mẩu tin: (2) Nếu sau khi xóa mẩu tin r mà L rỗng:

- Giải phóng L (xóa L).
- Xoá cặp khoá - con trỏ của L trong nút cha P của nó. *Nếu số con còn lại của P* $< \lceil m/2 \rceil$ thì xét nút P' **bên trái (hoặc bên phải) cùng mức** với P. **Nếu P' có ít nhất $\lceil m/2 \rceil + 1$ con (có dư)**: ***chuyển một con*** của P' sang P. Lúc này P và P' đều có ít nhất $\lceil m/2 \rceil$ con.
- Cập nhật lại giá trị khóa của P hoặc P' trong nút cha hoặc nút tổ tiên của chúng.

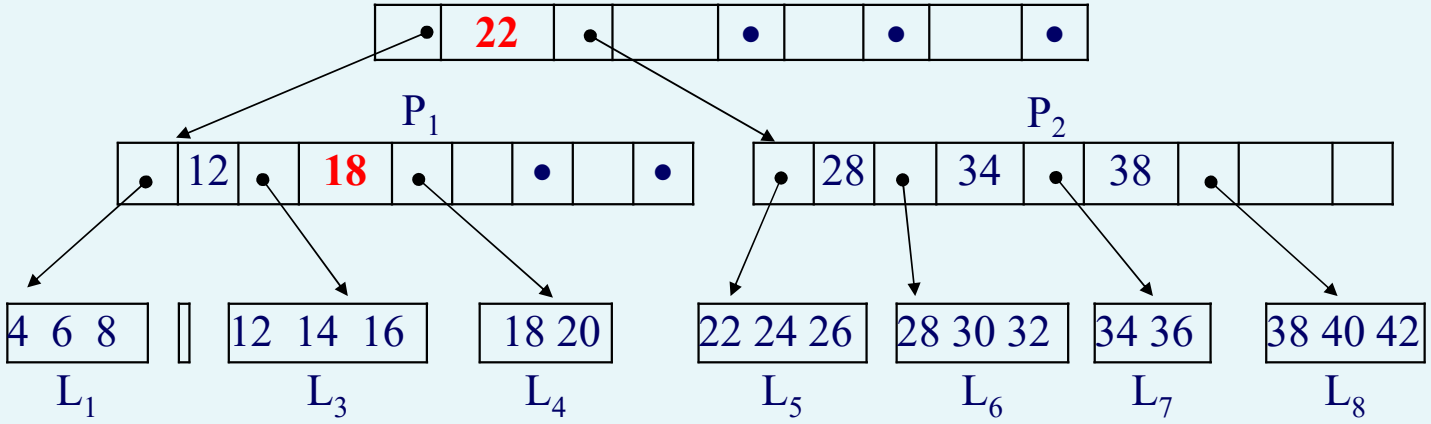
Ví dụ xóa mẩu tin (2)

Ví dụ 2 : Xóa mẩu tin r với khóa x = **10** trong tập tin được biểu diễn trong hình sau:



Ví dụ xóa mẩu tin (2)

Kết quả : Mẩu tin r với khóa x = **10** đã được xóa:



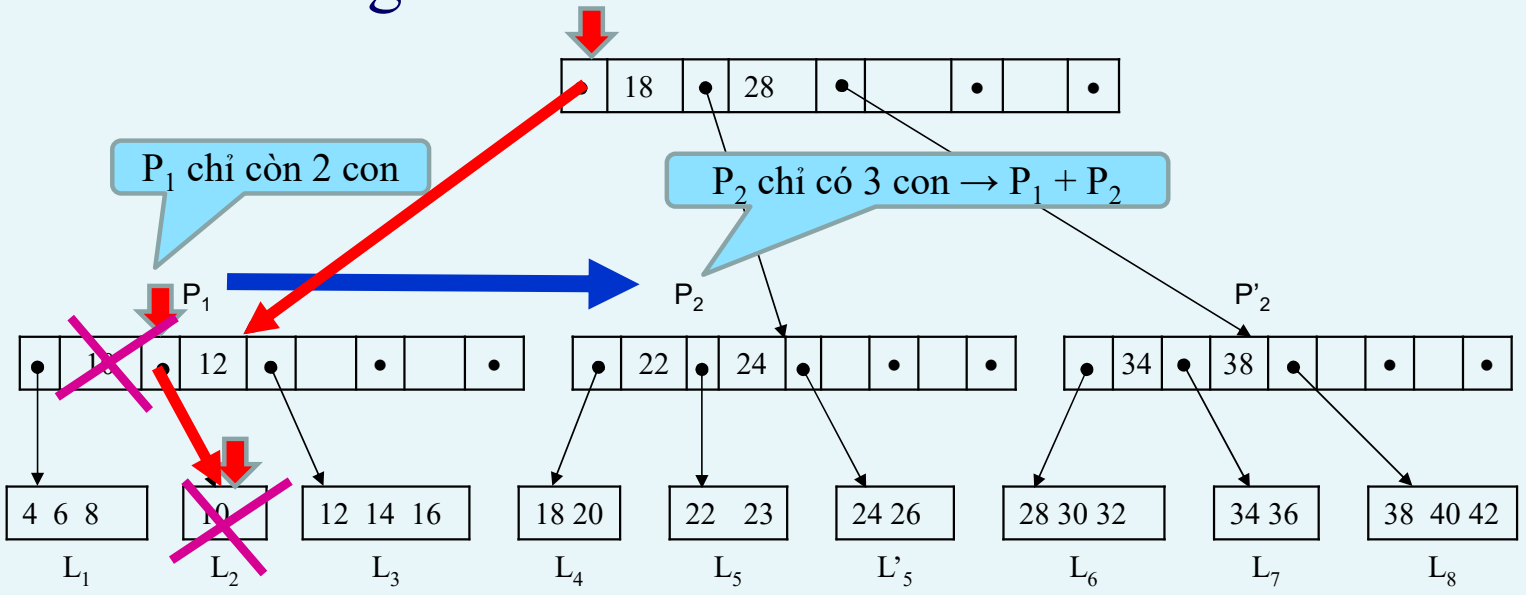
Tập tin B - cây : XÓA MẨU TIN

Xóa mẩu tin: (3) Nếu sau khi xóa mẩu tin r mà L rỗng:

- Giải phóng L (xóa L).
- Xoá cặp khoá - con trỏ của L trong nút cha P của nó. *Nếu số con còn lại của P* < ⌈m/2⌉ thì xét nút P' **bên trái (hoặc bên phải) cùng mức** với P. **Nếu P' có đúng ⌈m/2⌉ con (không có dư):** ***nối hai nút*** P' và P thành một nút có m con.
- Xóa **đệ quy** khóa và con trỏ P' trong cha của P'. Kết quả có thể dẫn tới việc nối 2 con của nút gốc tạo nên một gốc mới và giải phóng nút gốc cũ, độ cao của cây khi đó sẽ giảm 1.

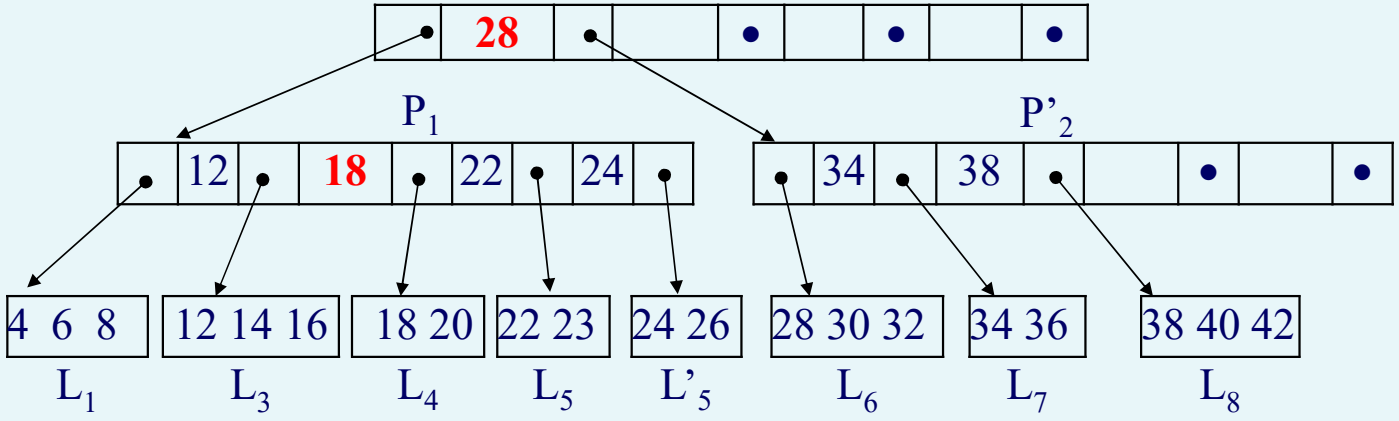
Ví dụ xóa mẩu tin (3)

Ví dụ 3 : Xóa mẩu tin r với khóa x = **10** trong tập tin được biểu diễn trong hình sau:



Ví dụ xóa mẩu tin (3)

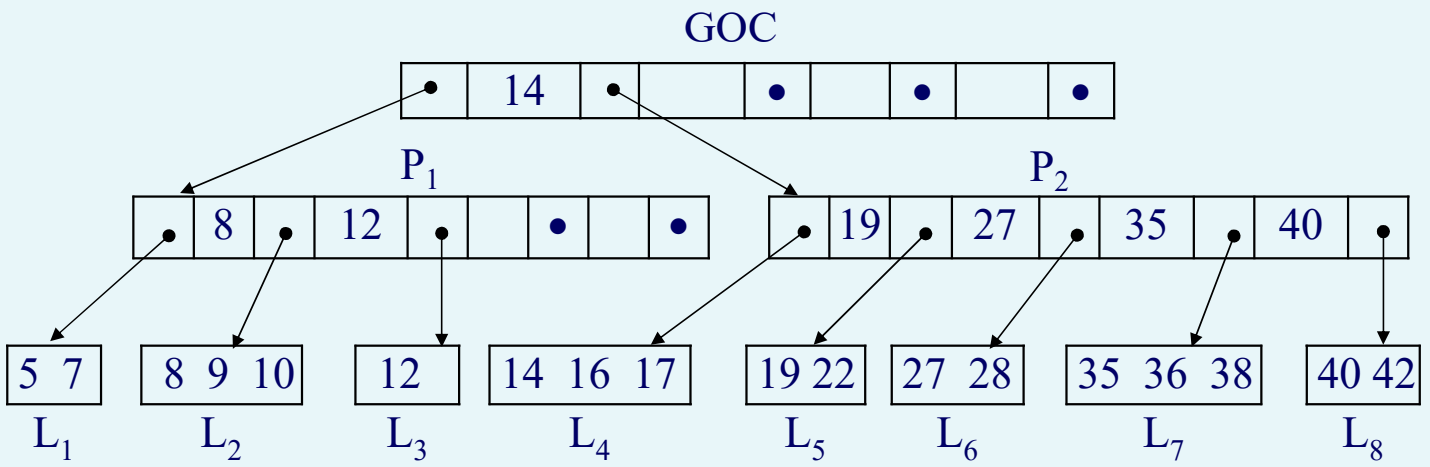
Kết quả : Mẩu tin r với khóa x = **10** đã được xóa:



Tập tin B – cây: Bài tập 1b

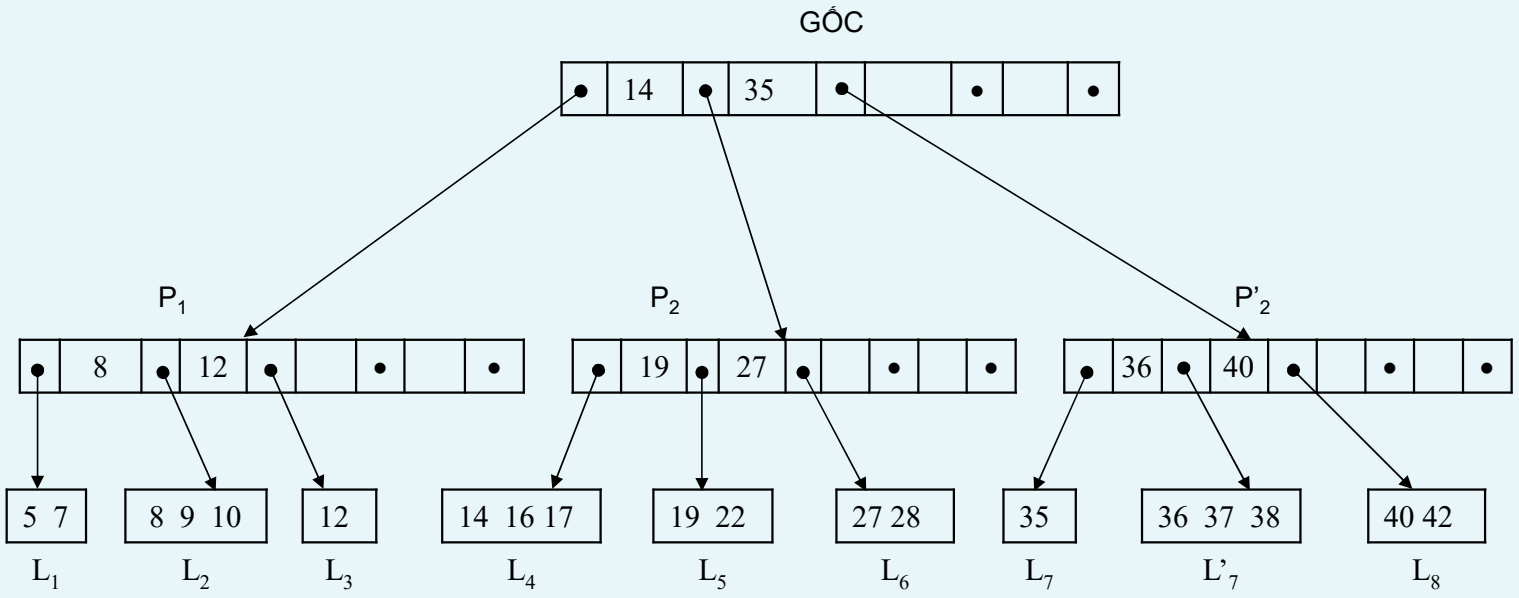
Bài tập: Cho tập tin bao gồm các mẩu tin với giá trị khóa là các số nguyên được tổ chức thành **B-cây bậc 5** với các nút lá chứa được nhiều nhất **3 mẩu tin** như sau:

1. b) : Xóa mẩu tin r với khóa x = **12** :



Tập tin B – cây: Bài tập 1c

1. c) : Xóa mẩu tin r với khóa x = **12** của tập tin kết quả câu a :

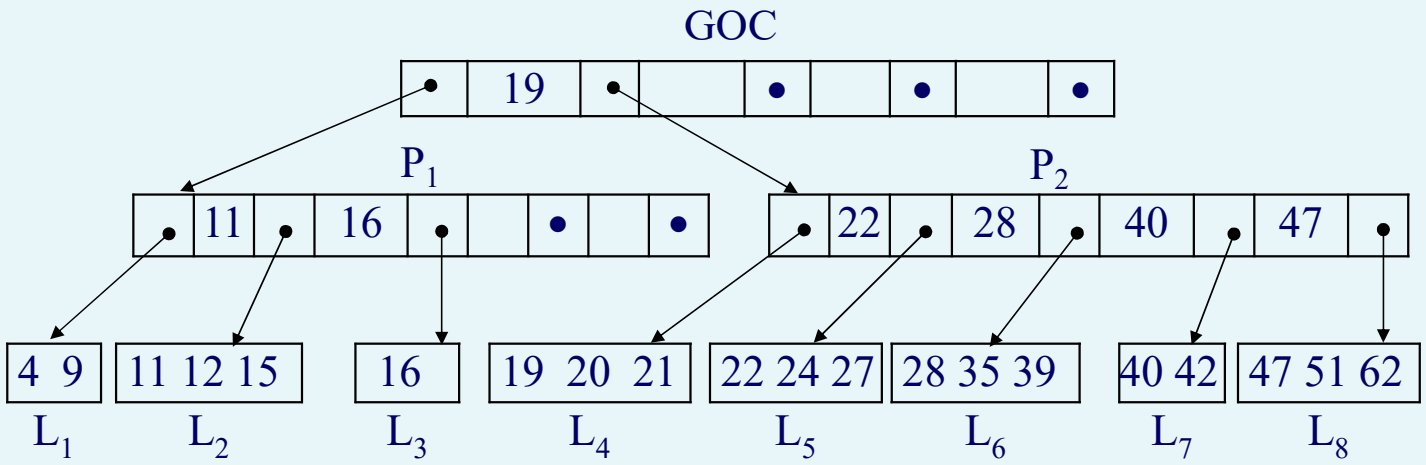


Tập tin B – cây: Bài tập

Bài tập: Cho tập tin gồm các mẩu tin giá trị khóa là các số nguyên được tổ chức thành **B-cây bậc 5** với nút lá chứa nhiều nhất **3 mẩu tin** như sau:

a) Thêm mẩu tin với khóa **29**

b) Xóa mẩu tin khóa **16** của B-cây kết quả câu a



Tập tin B – cây: Bài tập 2

Bài tập: Cho **B-cây bậc 3** với các nút lá chứa được nhiều nhất **2 mẩu tin** để tổ chức tập tin. Khởi đầu tập tin rỗng, hãy mô tả quá trình hình thành tập tin B-cây (bằng hình vẽ, sau mỗi thao tác vẽ một hình) khi thực hiện tuần tự các thao tác sau:

(1) Xen mẩu tin R có khóa **8**
(2) Xen mẩu tin R có khóa **2**
(3) Xen mẩu tin R có khóa **10**
(4) Xen mẩu tin R có khóa **1**
(5) Xen mẩu tin R có khóa **12**
(6) Xen mẩu tin R có khóa **3**
(7) Xen mẩu tin R có khóa **5**
(8) Xóa mẩu tin R có khóa **8**
(9) Xóa mẩu tin R có khóa **1**