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

NỘI DUNG

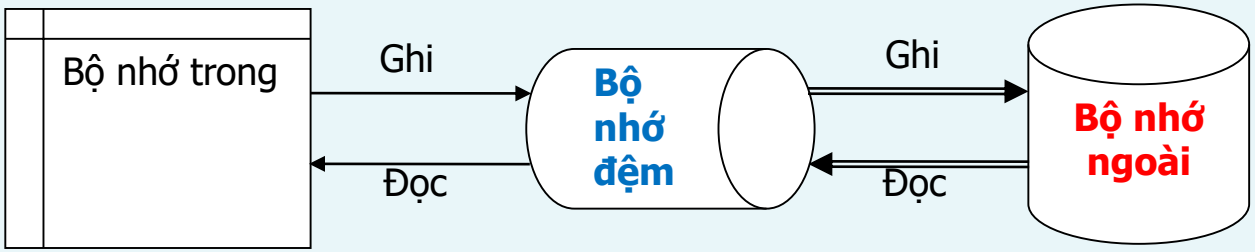
- Mô hình và đánh giá các xử lý ngoài.
- *Sắp xếp ngoài.*
- Lưu trữ thông tin trong tập tin:
  - Tập tin tuần tự
  - Tập tin bảng băm
  - Tập tin chỉ mục
  - **Tập tin B-cây**

Tại sao phải xử lý ngoài ?

- Trong các thuật toán đề cập trước đây, chúng ta đã giả sử rằng số lượng các dữ liệu đầu vào khá nhỏ để có thể chứa hết ở *bộ nhớ trong* (main memory).
- **Vấn đề**: Đối với các 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* ?
- Để có thể giải quyết các bài toán đó: phải 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 → Cần tìm các **cấu trúc dữ liệu** và **thuật toán** thích hợp cho việc xử lý dữ liệu lưu trữ trên *bộ nhớ ngoài* ?

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* là kiểu thích hợp nhất cho việc biểu diễn dữ liệu được 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.

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-cây bậc m** là cây tìm kiếm m-phân cân bằng có các tính chất sau:
  - **Nút gốc** hoặc là *lá* hoặc có *ít nhất 2 nút con*
  - **Mỗi nút**, trừ nút gốc và nút lá, có từ $\lceil m/2 \rceil$ đến m nút con
  - Các đường đi từ gốc tới lá có **cùng độ dài.**
  - Các khóa và cây con được **sắp xếp** theo cây tìm kiếm.

Khi nào dùng đến B - cây

- Khi cần hạn chế số thao tác đọc mỗi lần tìm kiếm trên cây

→ Thích hợp cho việc tìm kiếm trên bộ nhớ ngoài.

- Nếu dùng cây tìm kiếm **nhị phân n nút**, cần trung bình **logn** phép truy xuất khối để tìm kiếm mẩu tin → Nếu dùng cây tìm kiếm **m phân (B-cây)** thì chỉ cần $\log_m n$ phép truy xuất khối.
- Chiều cao cây = $\log_m n$: Nếu tăng m → chiều cao cây giảm rất nhanh.

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.
- Giả sử mỗi nút lá lưu trữ được nhiều nhất **b mẩu tin.**
- 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 $\mathbf{p_i}$ $(0 \le i \le n)$ là con trỏ, trỏ tới nút con thứ i của nút đó và $\mathbf{k_i}$ là các giá trị khóa.

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

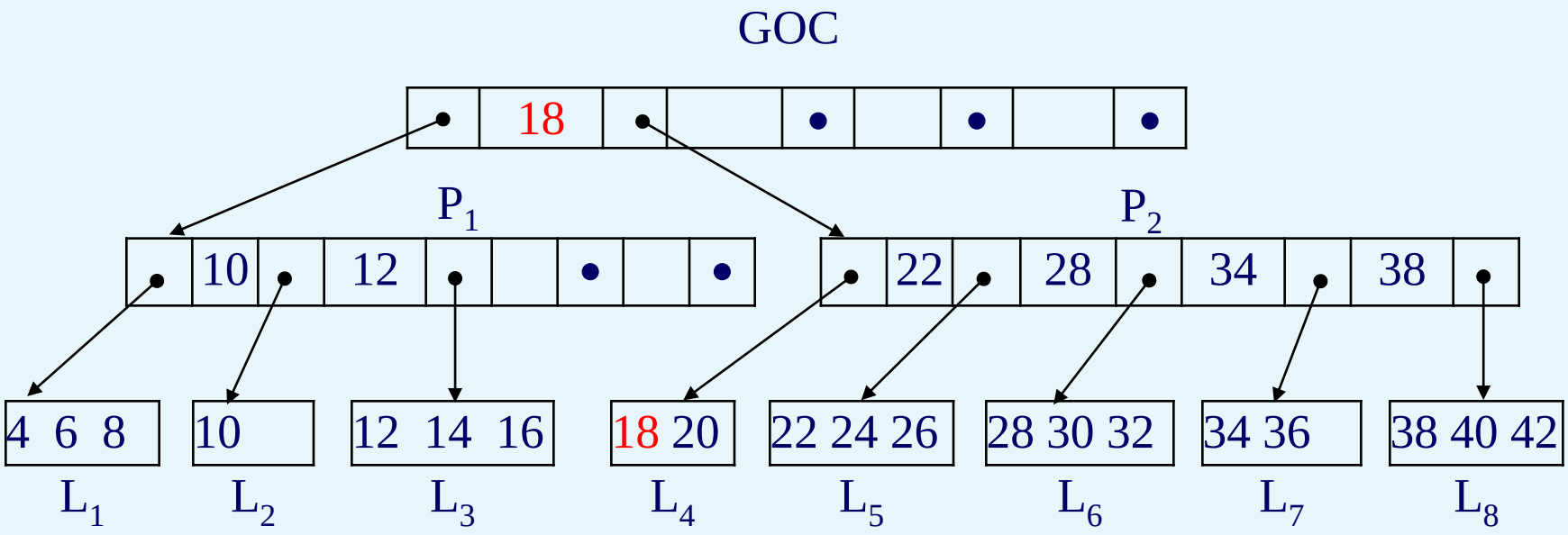
- Tất cả các khoá trong cây con được trỏ bởi p_0 đều < $\mathbf{k_1}$.

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

- Tất cả các khoá nằm trong cây con được trỏ bởi p_n đều $\ge \mathbf{k_n}$.

Tập tin B – cây: Ví dụ

Ví dụ: Cho tập tin bao gồm 20 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**.

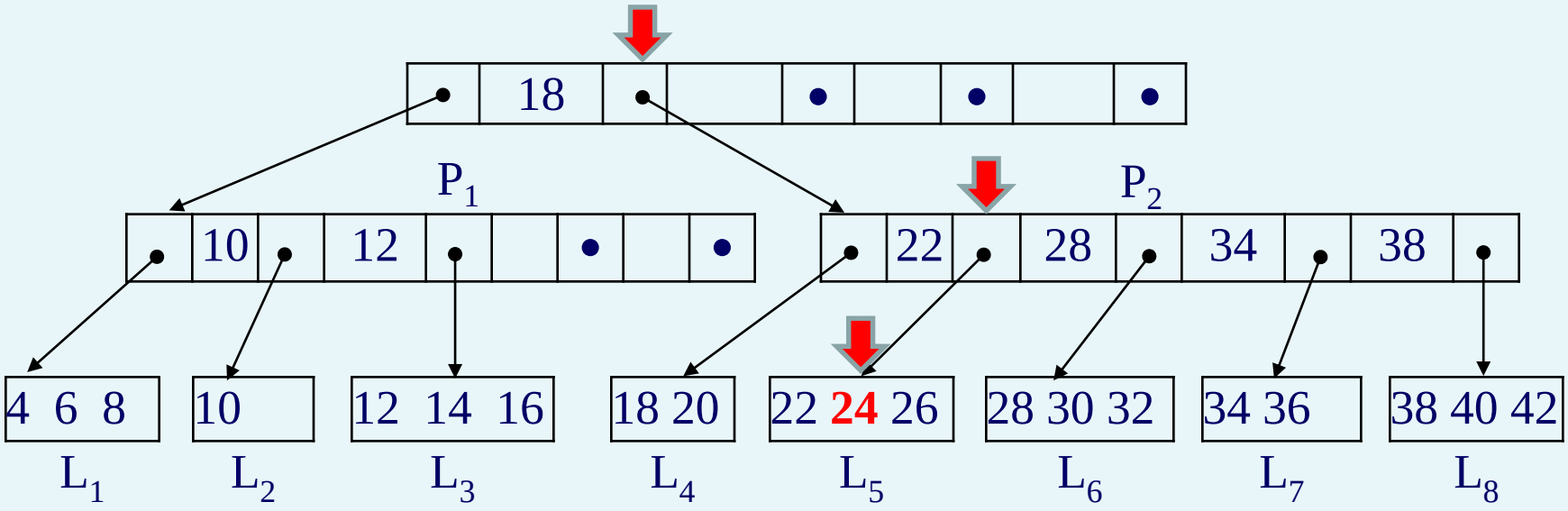


Tập tin B - cây : Các thao tác

- **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 $(p_0,k_1,p_1,k_2,p_2, \dots ,k_n,p_n)$ 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 một 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 : Các thao tác

- **Thêm mẩu tin:** Tìm r. Việc tìm kiếm này sẽ 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.

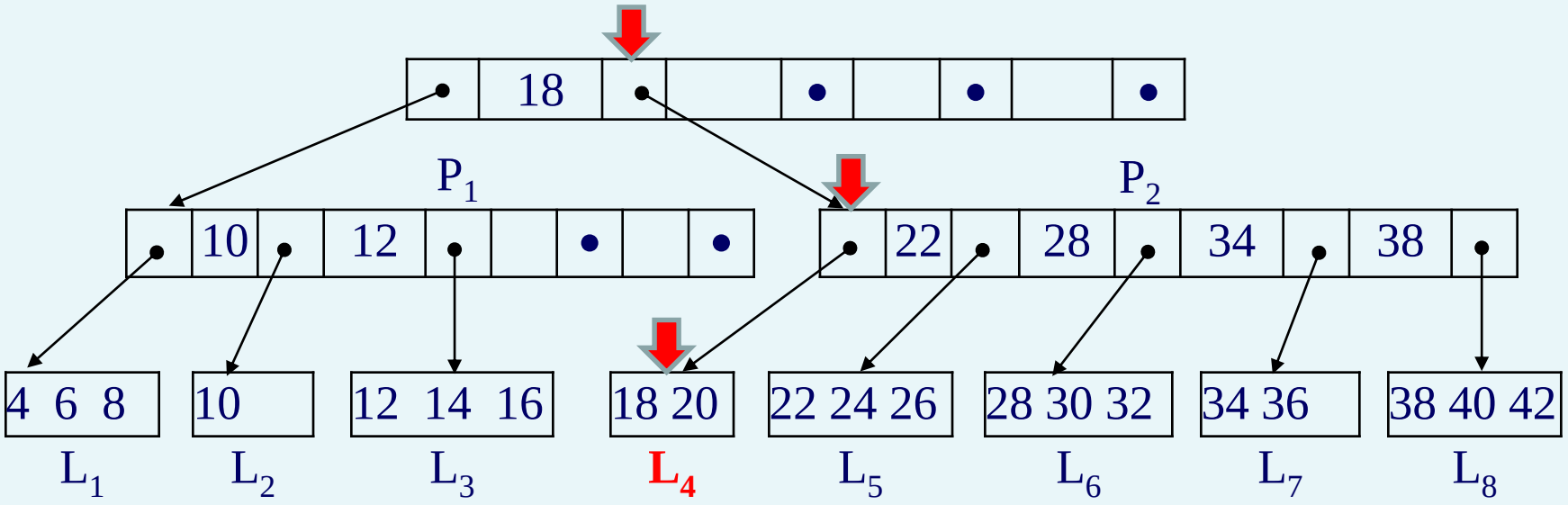
    (1) **Nếu L còn chỗ**: thêm r vào đúng thứ tự và kết thúc.

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

- Giả sử nút P là cha của L: Xen đệ quy vào P cặp khóa k' - con trỏ p' tương ứng của L'.
- Trường hợp trước khi xen k' và p', P đã có đủ m con thì cấp thêm khối mới P', chuyển một số con của P sang P' và xen con mới vào P hoặc P' sao cho cả P và P' đều có ít nhất $\lceil m/2 \rceil$ con. Việc chia cắt P $\Rightarrow$ phải xen cặp khóa-con trỏ vào nút cha của P, ... Quá trình này có thể dẫn tới nút gốc và cũng có thể phải chia cắt nút gốc, trong trường hợp này phải 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.

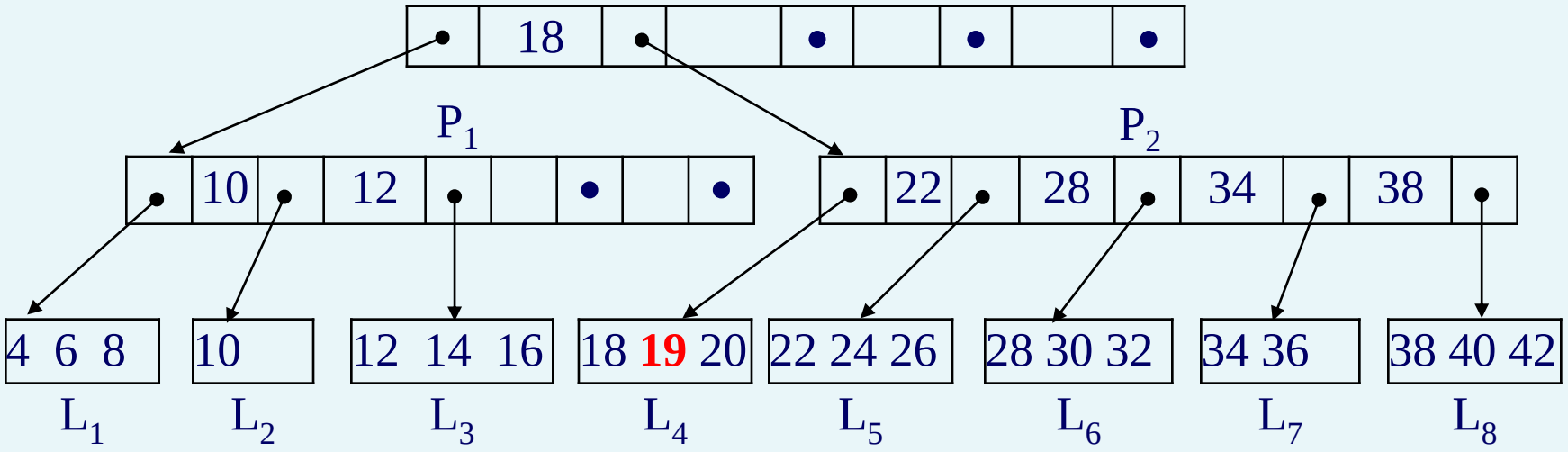
Ví dụ thêm mẩu tin mới (1)

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



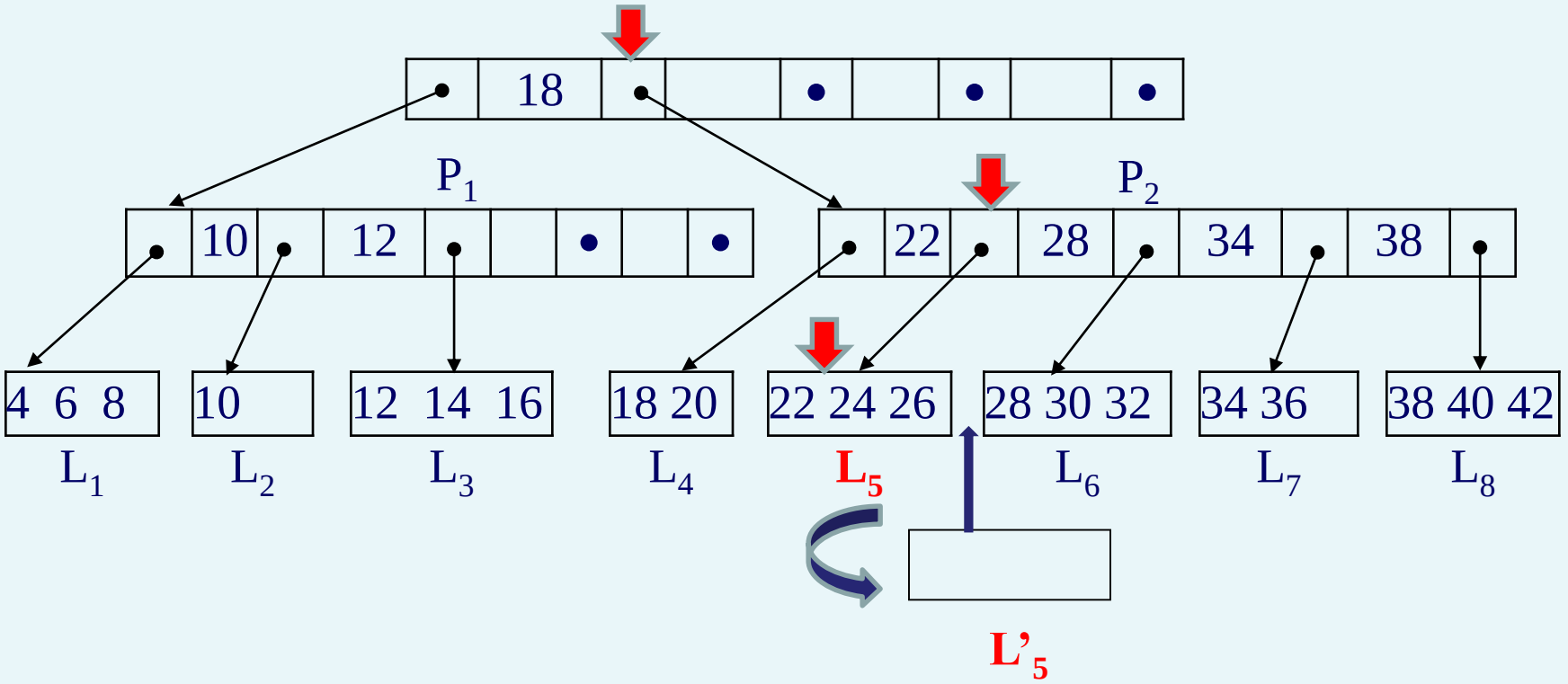
Ví dụ thêm mẩu tin mới (1)

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



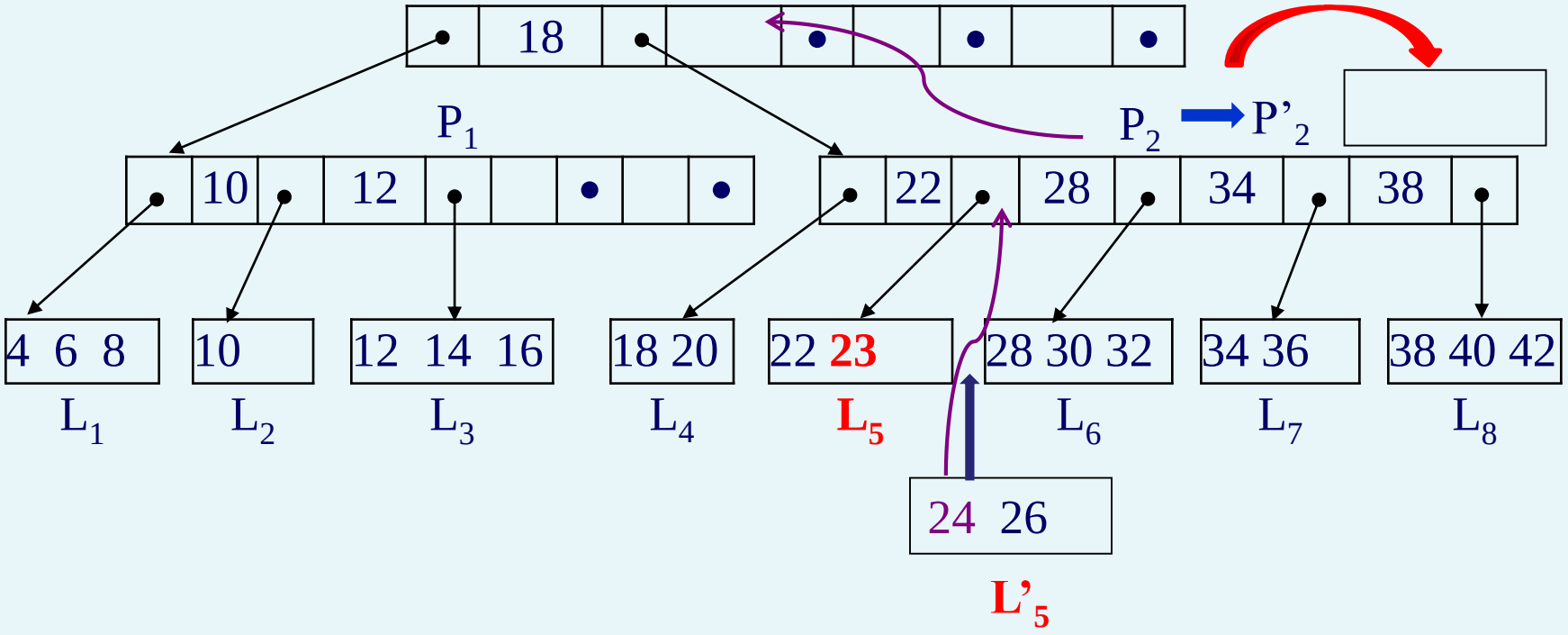
Ví dụ thêm mẩu tin mới (2)

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



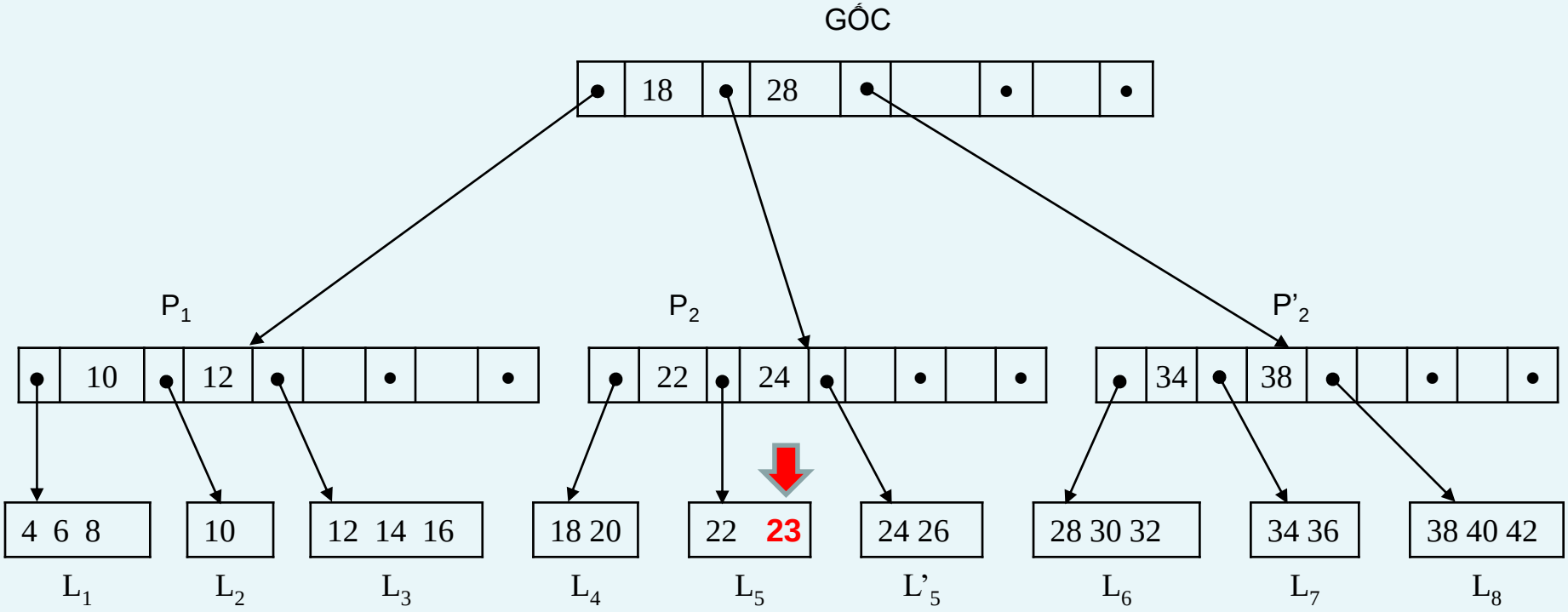
Ví dụ thêm mẩu tin mới (2)

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



Ví dụ thêm mẩu tin mới (2)

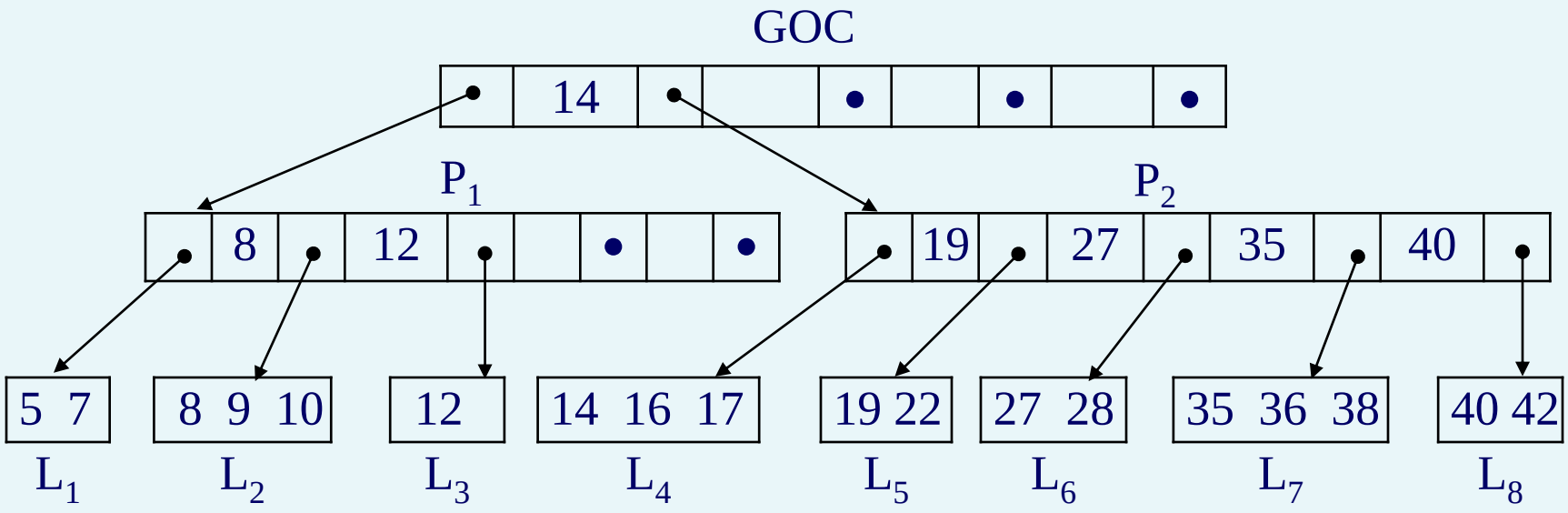
Kết quả : Mẩu tin r với khóa x = **23** đã được thêm vào:



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: Bài giải 1a

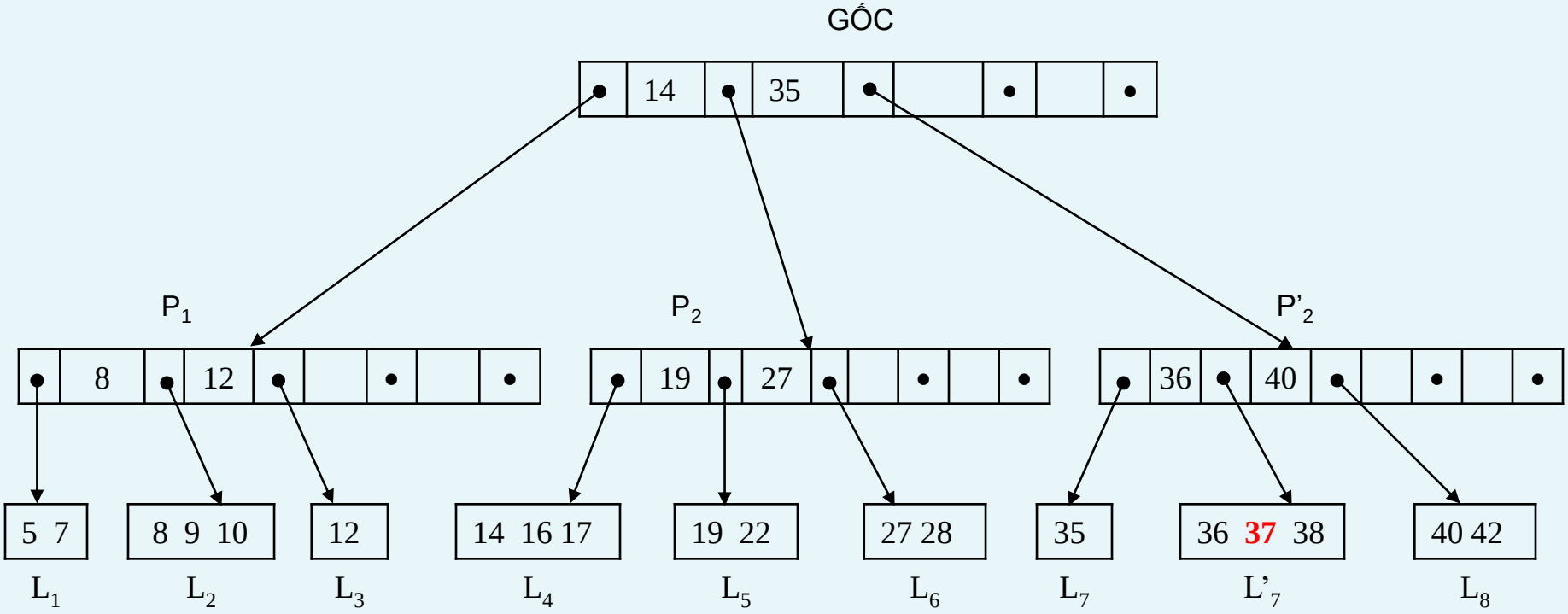
1. **a)** : Thêm mẩu tin r với khóa x = **37** :

- Quá trình tìm kiếm đi từ nút GỐC, xuống P_2, tới lá L_7.
- **Xen r vào L_7**: Vì L_7 đã đủ 3 mẩu tin, nên yêu cầu cấp phát nút lá mới L'_7, chuyển $\lceil b/2 \rceil = 2$ mẩu tin cuối (khóa 36, 38) sang L'_7, sau đó xen r (x=37) vào L'_7 .
- **Xen L'_7 vào P_2**: Vì P_2 đã có đủ 5 con, nên yêu cầu cấp phát nút trong mới P'_2, chuyển $\lceil m/2 \rceil = 3$ nút lá cuối (L_7, L'_7 và L_8) sang P'_2 và xen L'_7 vào P'_2.
- **Xen P'_2 vào GỐC**: Vì nút gốc còn chỗ nên xen khóa đầu L_7 (khóa 35) và con trỏ của P'_2 vào.

Cập nhật lại các khóa, ta được B-cây như sau:

Tập tin B – cây: Bài giải 1a

Kết quả : Mẩu tin r với khóa x = **37** đã được thêm vào:



Tập tin B - cây : Các thao tác

- **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.

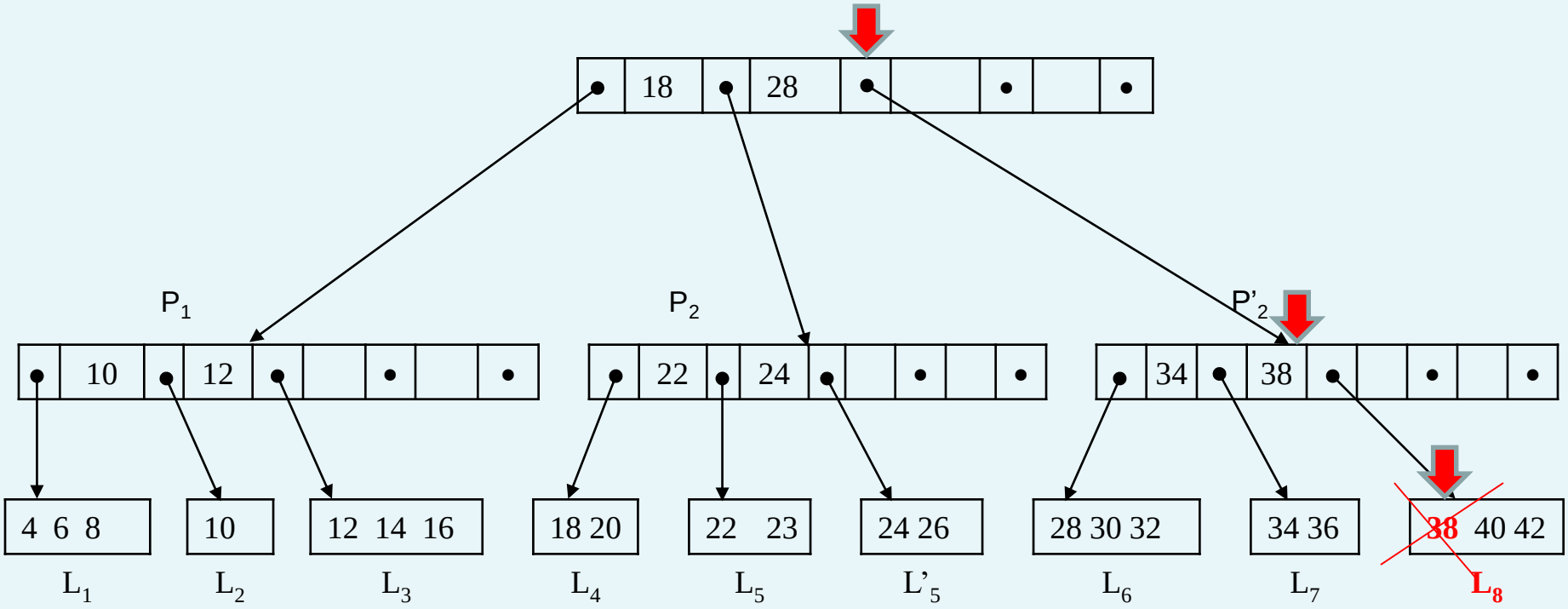
Có 3 trường hợp cần xét :

(1) Nếu r là mẩu tin đầu tiên của L, thì phải quay lui lên nút P là cha của L để đặt lại giá trị khóa của L trong P, giá trị mới này bằng giá trị khóa của mẩu tin mới đầu tiên của L.

Trong trường hợp L là con đầu tiên của P, thì khóa của L không nằm trong P mà nằm trong tổ tiên của P, chúng ta phải quay lui lên mà sửa đổi.

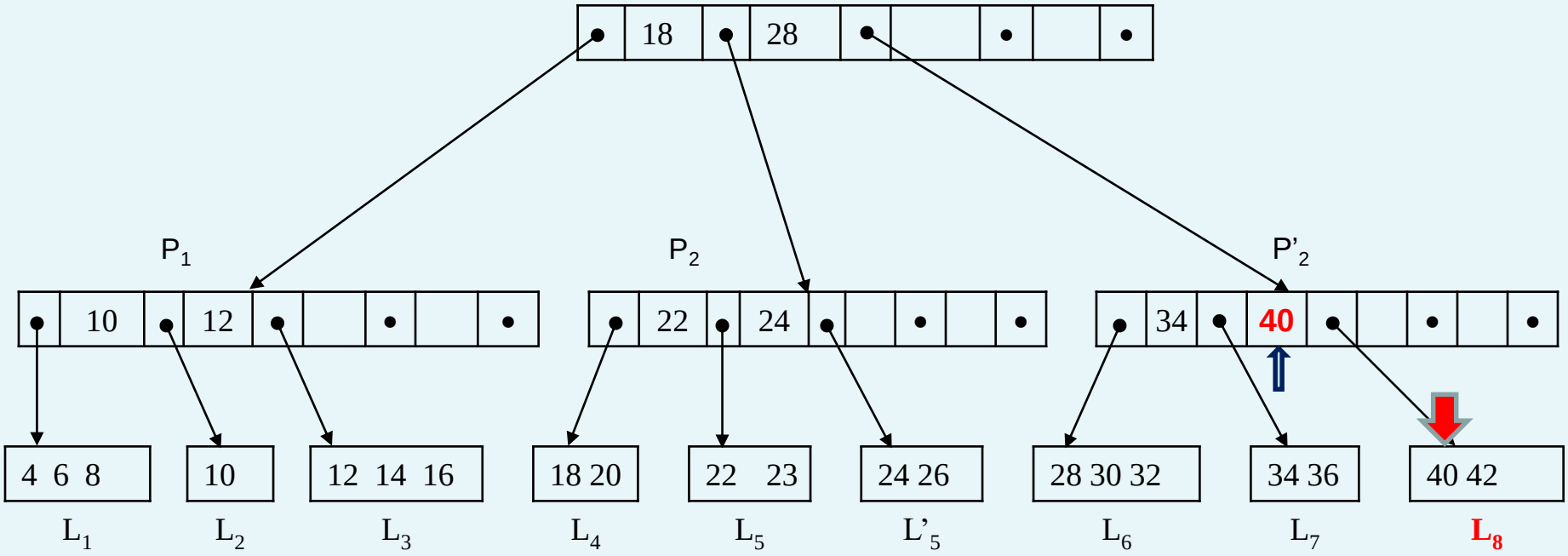
Ví dụ xóa mẩu tin (1)

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 : Các thao tác

- **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",

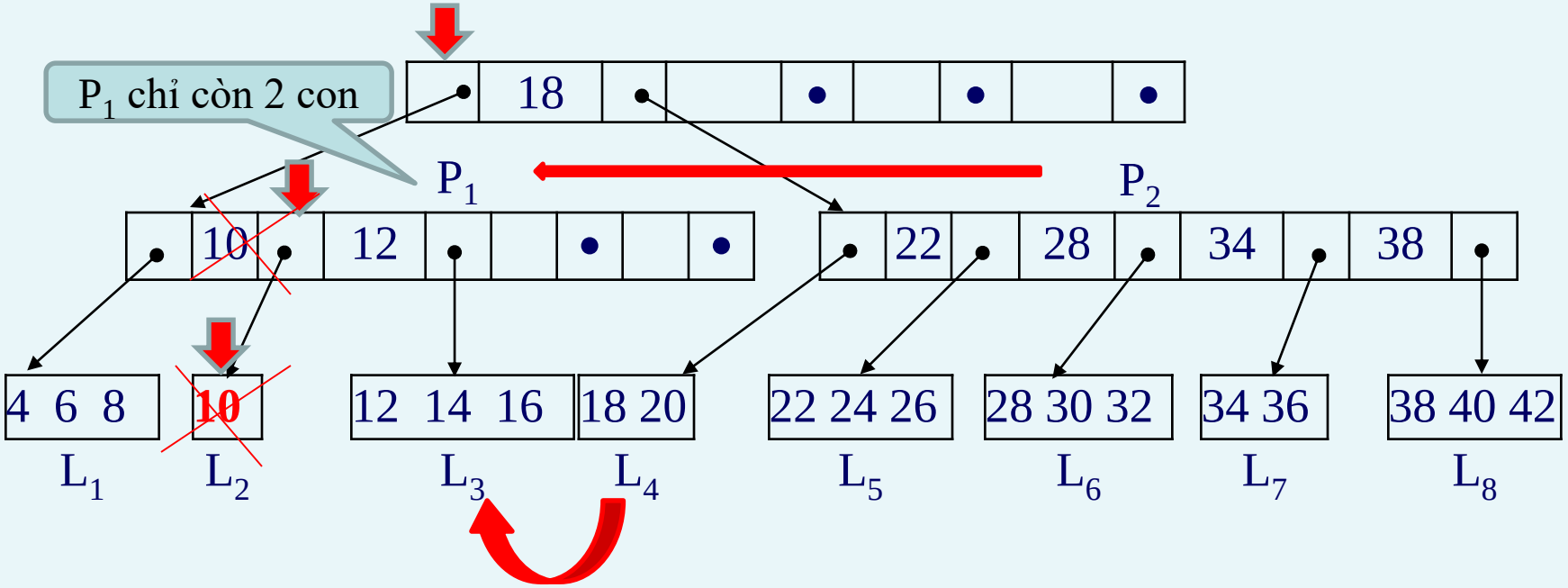
  - *Ngược lại* thì L là nút lá có thể xóa r trong đó.

Có 3 trường hợp cần xét :

(2) Nếu sau khi xóa mẩu tin r mà L trở nên rỗng thì giải phóng L và quay lui lên nút P là cha của L để xoá cặp khoá-con trỏ của L trong P. Nếu số con của P bây giờ (sau khi xoá khóa-con trỏ của L) < $\lceil$ **m/2** $\rceil$ thì kiểm tra nút P' ngay **bên trái hoặc bên phải** và cùng mức với P. Nếu P' có ít nhất $\lceil m/2 \rceil$+1 con, chuyển một con của P' sang P. Lúc này cả P và P' có ít nhất $\lceil m/2 \rceil$. Sau đó, phải cập nhật lại giá trị khóa của P hoặc P' trong cha của chúng, và nếu cần phải sửa cả trong tổ tiên của chúng.

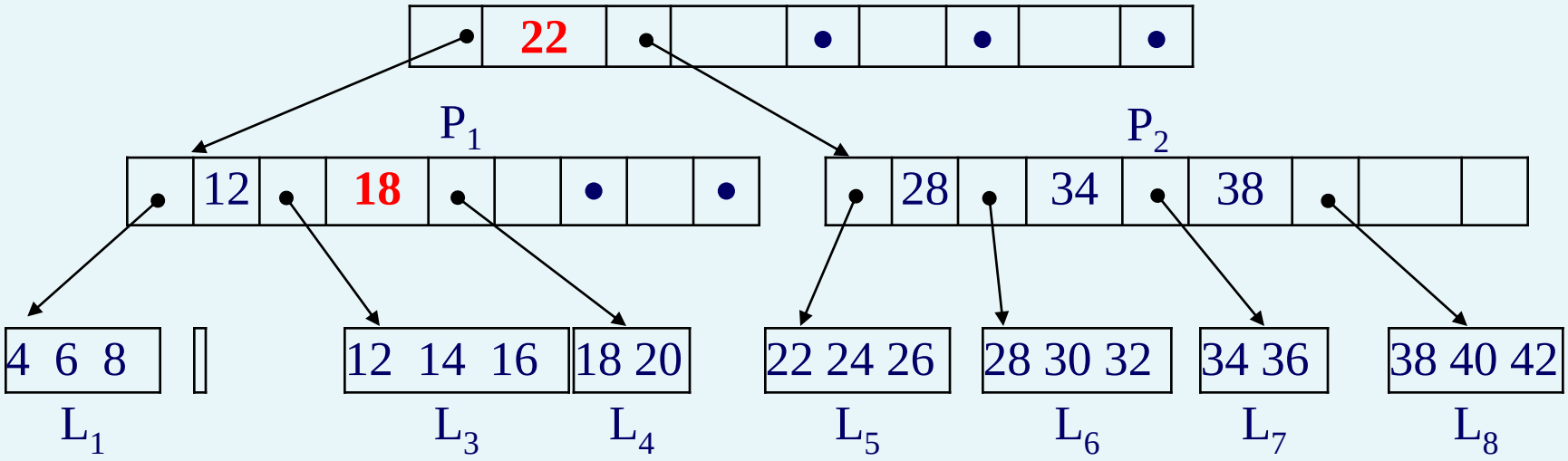
Ví dụ xóa mẩu tin (2)

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 : Các thao tác

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.

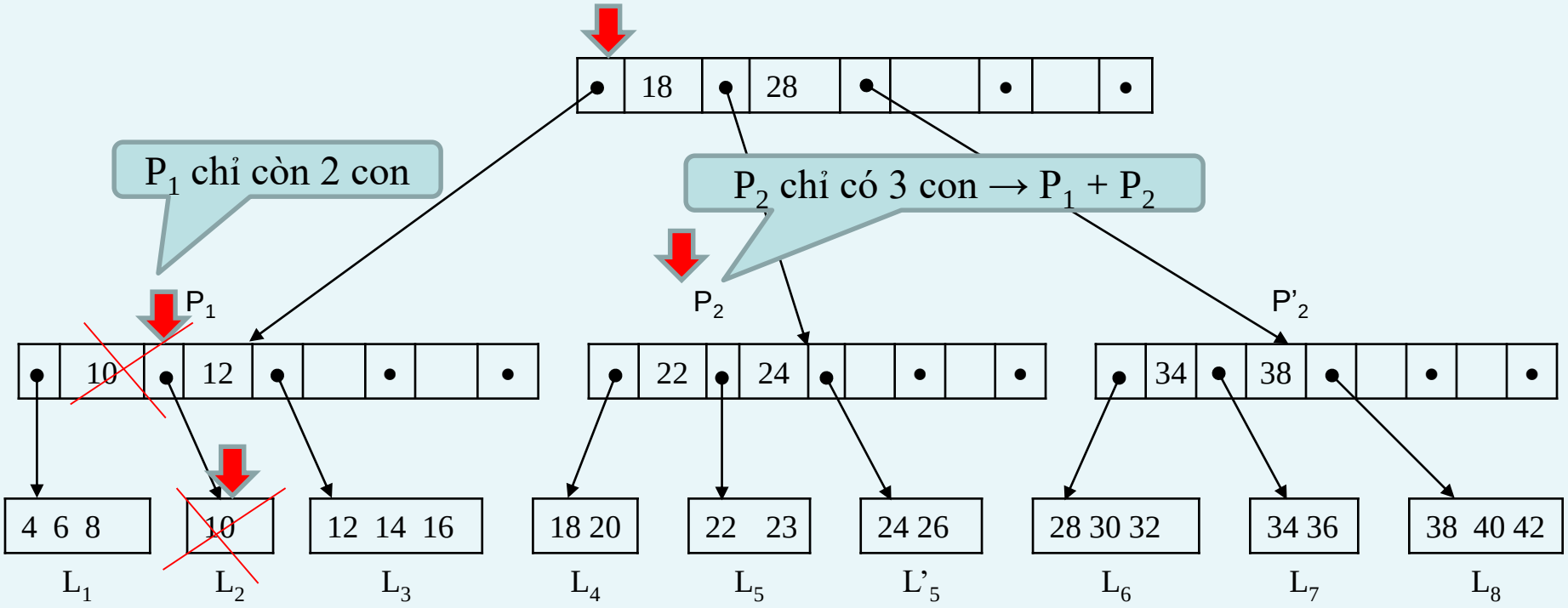
Có 3 trường hợp cần xét :

(3) Nếu P' có đúng $\lceil m/2 \rceil$ con, ta nối P và P' thành một nút có đúng m con. Sau đó ta phải xóa khóa và con trỏ của P' trong nút cha của P'.

Việc xóa này có thể phải quay lui lên tổ tiên của P'. Kết quả của quá trình xóa đệ quy này có thể dẫn tới việc nối hai con của nút gốc, tạo nên một gốc mới và giải phóng nút gốc cũ, độ cao của cây khi đó sẽ giảm đi 1.

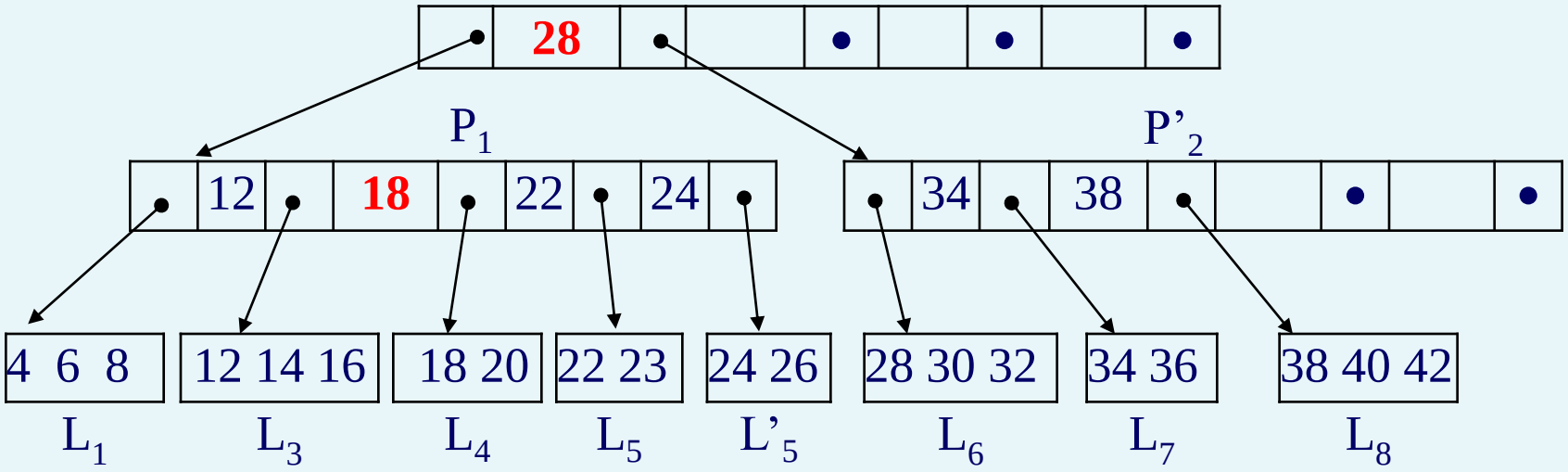
Ví dụ xóa mẩu tin (3)

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)

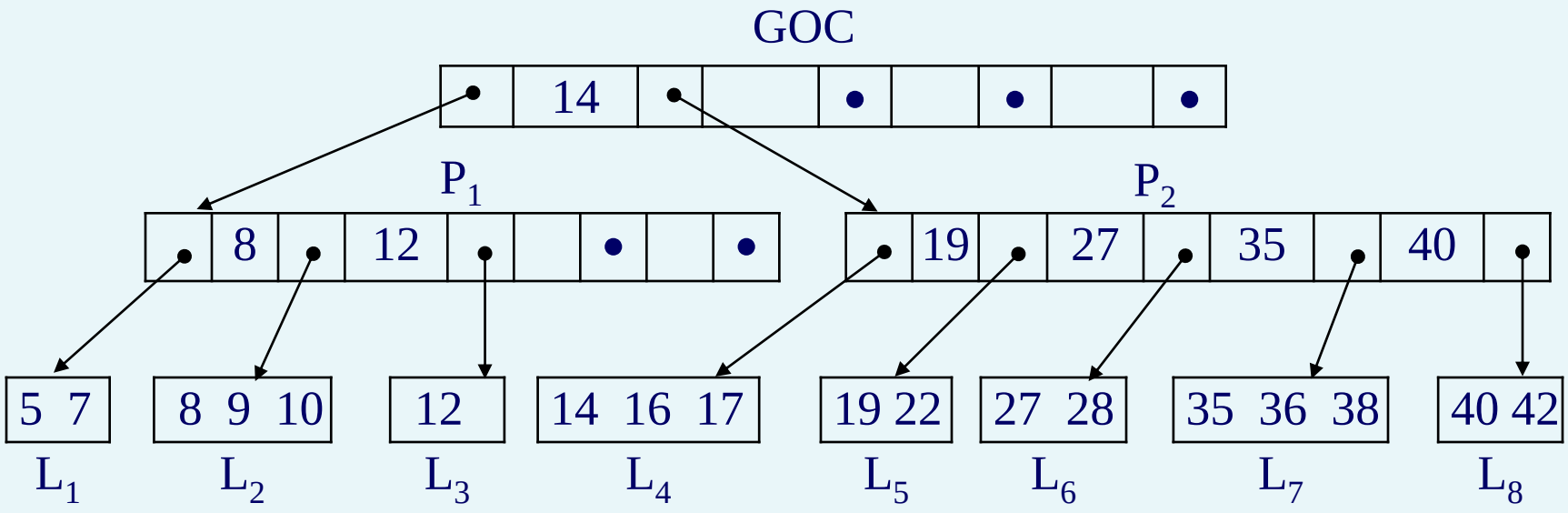
Kết quả : Mẩu tin r với khóa x = **10** đã được xóa:



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 giải 1b

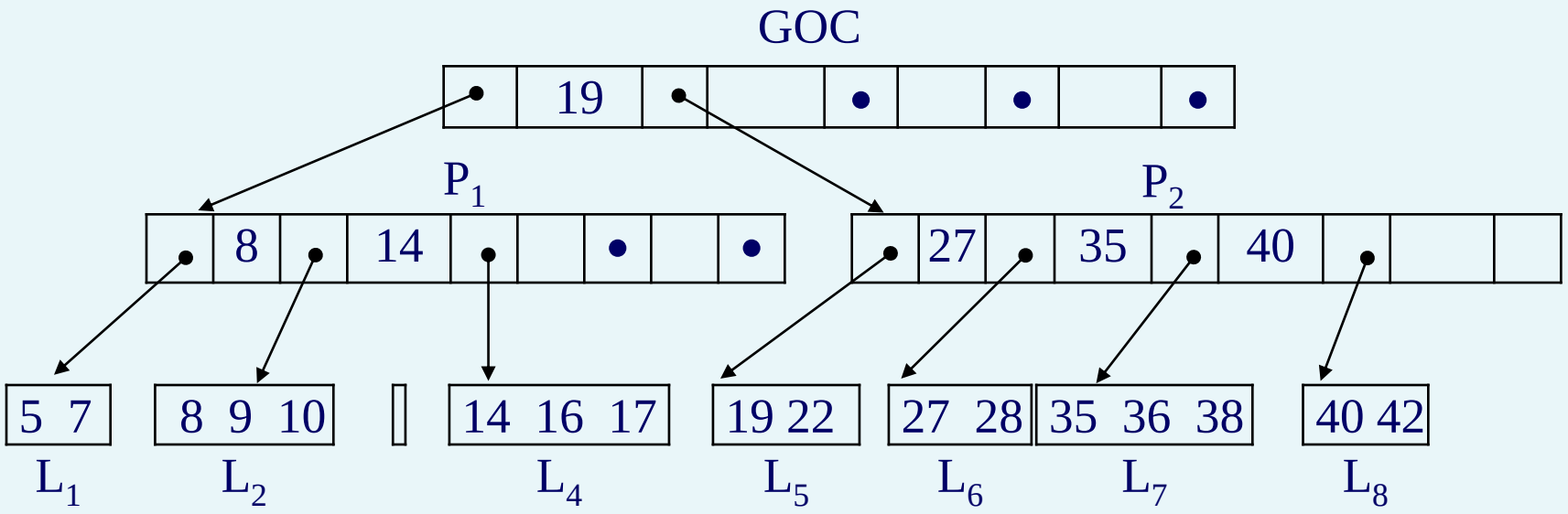
1. **b)** : Xóa mẩu tin r với khóa x = **12** :

- Quá trình tìm kiếm đi từ GỐC, xuống P_1 và tới lá L_3.
- **Xóa mẩu tin r (khóa 12) khỏi L_3**. L_3 rỗng, giải phóng L_3.
- **Xóa khóa 12 và con trỏ của L_3 trong P_1**. Lúc này, P_1 chỉ còn 2 con ($< \lceil m/2 \rceil = 3$).
- Xét P_2, bên phải cùng mức với P_1, P_2 có 5 con ($> \lceil m/2 \rceil = 3$) nên ta chuyển một con trái nhất (lá L_4) từ P_2 sang P_1,

Cập nhật lại khoá của P_2 trong nút GỐC, ta được B-cây như sau:

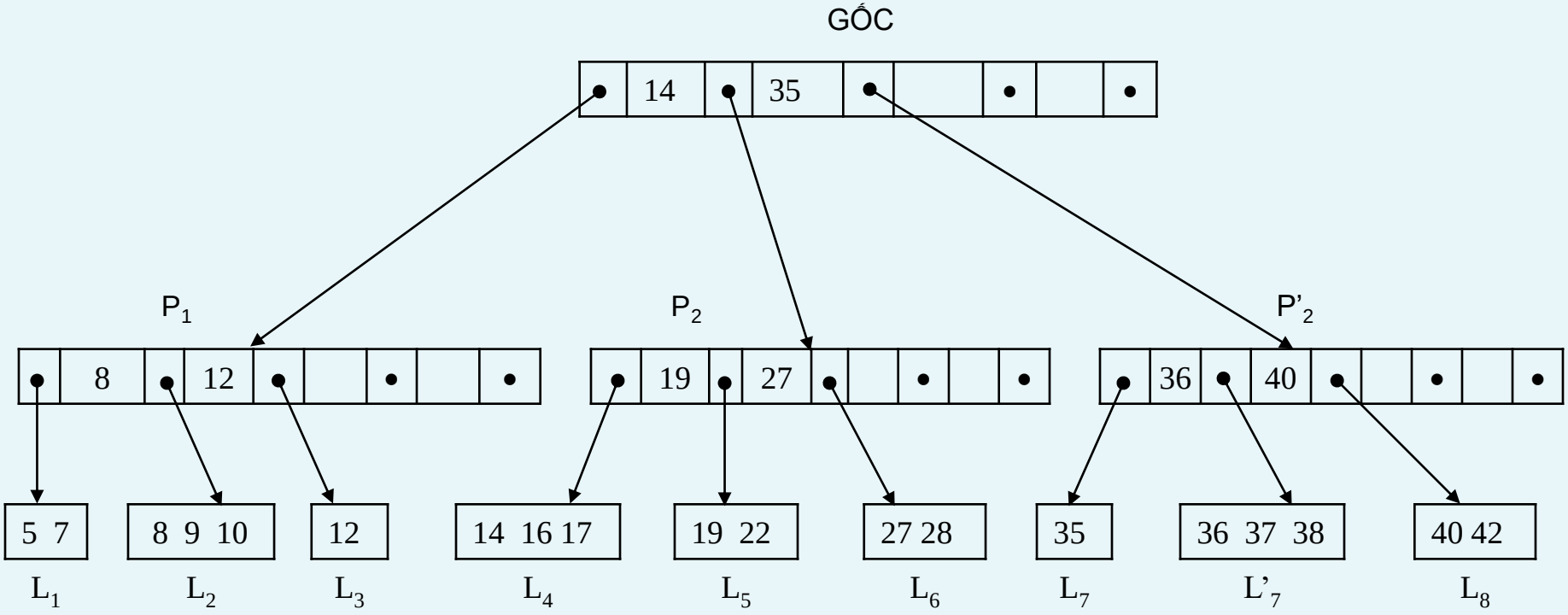
Tập tin B – cây: Bài giải 1b

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 giải 1c

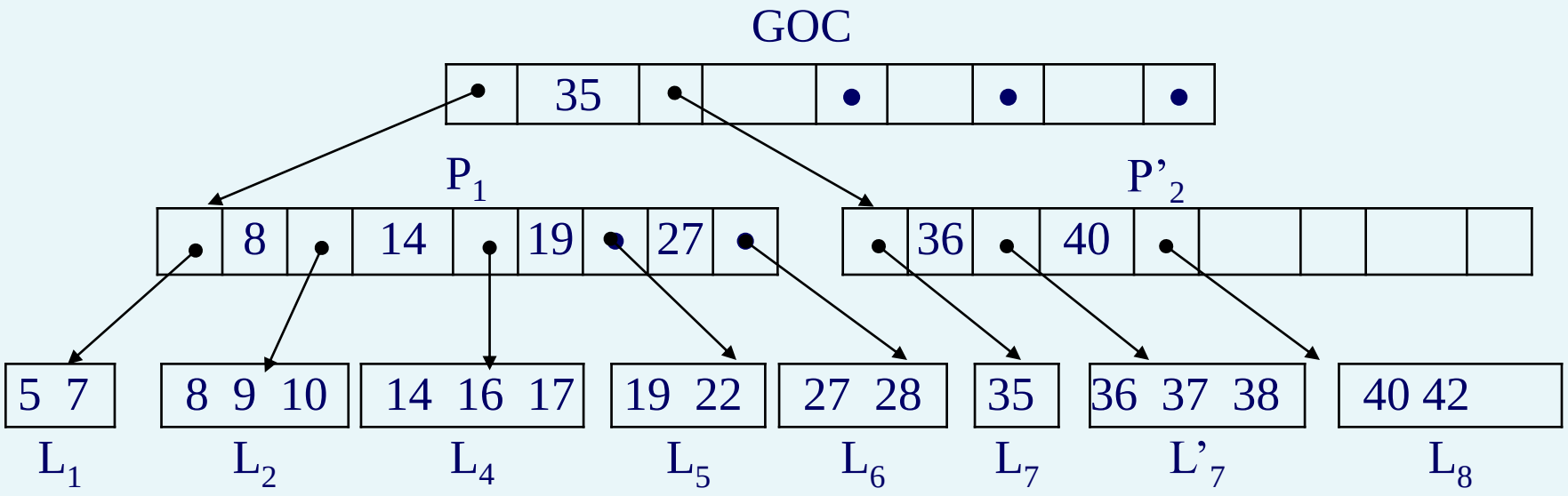
1. **c)** : Xóa mẩu tin r với khóa x = **12** :

- Quá trình tìm kiếm đi từ GỐC, xuống P_1 và tới lá L_3.
- **Xóa mẩu tin r (khóa 12) khỏi L_3**. L_3 rỗng, giải phóng L_3.
- **Xóa khóa 12 và con trỏ của L_3 trong P_1**. Lúc này, P_1 chỉ còn 2 con ($< \lceil m/2 \rceil = 3$).
- Xét P_2 , bên phải cùng mức với P_1, vì P_2 có đúng $\lceil m/2 \rceil = 3$ con nên ta nối P_2 vào P_1, giải phóng P_2 bằng cách xóa khóa và con trỏ của P_2 trong nút GỐC,

Cập nhật lại khóa, ta được B-cây như sau:

Tập tin B – cây: Bài giải 1c

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



Tập tin B – cây: Bài tập 2

Bài tập: Giả sử **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:

Xen mẩu tin R có khóa **8**

Xen mẩu tin R có khóa **2**

Xen mẩu tin R có khóa **10**

Xen mẩu tin R có khóa **1**

Xen mẩu tin R có khóa **12**

Xen mẩu tin R có khóa **3**

Xen mẩu tin R có khóa **5**

Xóa mẩu tin R có khóa **8**

Xóa mẩu tin R có khóa **1**

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:

a) Thêm mẩu tin với khóa **29**

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