TRƯỜNG ĐẠI HỌC AN GIANG
KHOA CÔNG NGHỆ THÔNG TIN

PHÂN TÍCH THIẾT KẾ GIẢI THUẬT

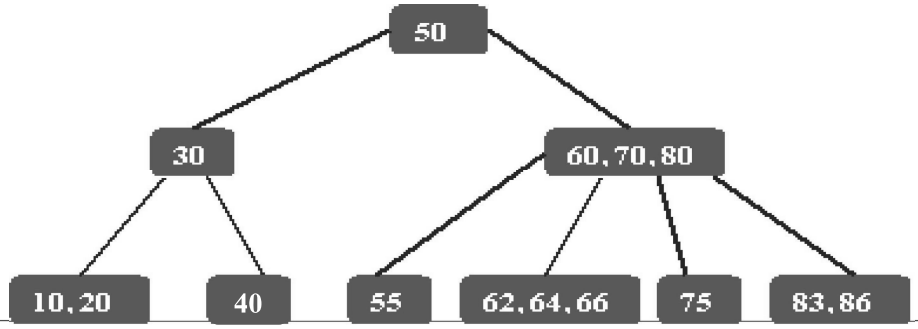
Giảng viên phụ trách:
NGUYỄN THÁI DƯ

Chương 5 CÂY 2-3-4

- Giới thiệu về cây 2-3-4
- Tổ chức
- Tìm kiếm
- Thêm vào

Giới thiệu

- Đối với cây nhị phân, mỗi nút chỉ có một mục dữ liệu và có thể có hai nút con.
- Nếu chúng ta cho phép một nút có nhiều mục dữ liệu và nhiều nút con thì kết quả là ta được cây nhiều nhánh



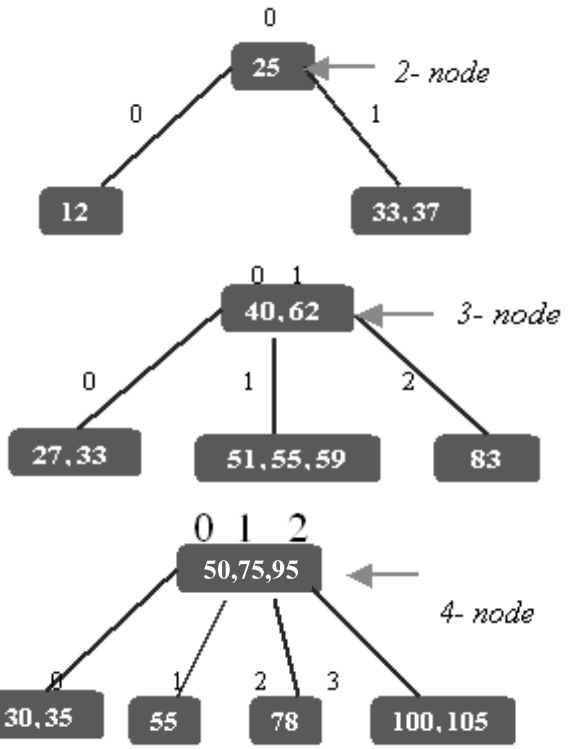
Giới thiệu về cây 2-3-4

- **Cây 2-3-4** là cây nhiều nhánh mà mỗi nút của nó có thể có đến bốn nút con và ba mục dữ liệu.
- Các số **2, 3** và **4** trong cụm từ cây 2-3-4 có ý nghĩa là khả năng có bao nhiêu liên kết đến các node con có thể có được trong một node cho trước.

Giới thiệu về cây 2-3-4 (tt)

- Đối với các node không phải là lá, có 3 cách sắp xếp sau:
  - Một node với một mục dữ liệu thì luôn luôn có 2 con.
  - Một node với hai mục dữ liệu thì luôn luôn có 3 con.
  - Một node với ba mục dữ liệu thì luôn luôn có 4 con.
- Nói cách khác, nếu số con là L và số mục dữ liệu là D, thì:

  $L = D + 1$



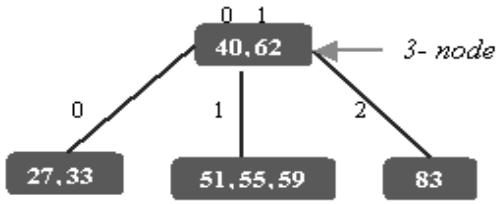
Giới thiệu về cây 2-3-4 (tt)

- Một node lá thì không có node con nhưng có thể chứa 1, 2 hoặc 3 mục dữ liệu.
- Trong ***cây 2-3-4*** không tồn tại node chỉ có liên kết đơn. Một node với 1 mục dữ liệu luôn luôn phải có 2 liên kết, trừ khi nó là node lá (node không có liên kết nào).

Tổ chức cây 2-3-4

- Các mục dữ liệu trong mỗi node được sắp xếp theo thứ tự tăng dần từ trái sang phải (sắp xếp từ thấp đến cao).
- Trong cây tìm kiếm nhị phân: **NL<NRoot<NR** *khoá của nút cây con bên trái nhỏ hơn khóa của nút đang xét và khóa của nút đang xét nhỏ hơn khoá của nút cây con bên phải.*
- ***Cây 2-3-4*** cũng có tính chất như trên, nhưng có thêm đặc điểm sau:
  - Tất cả các nút con của cây con có gốc thứ i thì có các giá trị khoá nhỏ hơn khoá i.

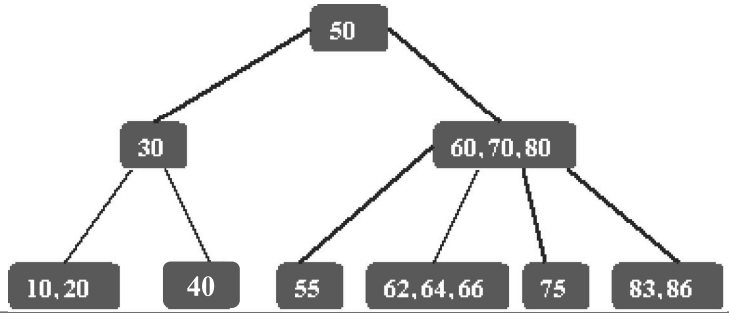
Tổ chức cây 2-3-4



- Trong cây 2-3-4 thì nguyên tắc cũng giống như cây tìm kiếm nhị phân, nhưng có thêm một số điểm sau:
  - Tất cả các node con của cây con có gốc tại node con thứ 0 thì có các giá trị khoá nhỏ hơn khoá 0.
  - Tất cả các node con của cây con có gốc tại node con thứ 1 thì có các giá trị khoá lớn hơn khoá 0 và nhỏ hơn khoá 1.
  - Tất cả các node con của cây con có gốc tại node con thứ 2 thì có các giá trị khoá lớn hơn khoá 1 và nhỏ hơn khoá 2.
  - Tất cả các node con của cây con có gốc tại node con thứ 3 thì có các giá trị khoá lớn hơn khoá 2.

Tìm kiếm

- Thao tác tìm kiếm trong cây 2-3-4 tương tự như thủ tục tìm kiếm trong cây nhị phân.Việc tìm kiếm bắt đầu từ node gốc và chọn liên kết dẫn đến cây con với phạm vi giá trị phù hợp.
- Ví dụ: tìm kiếm mục dữ liệu với khoá là 64



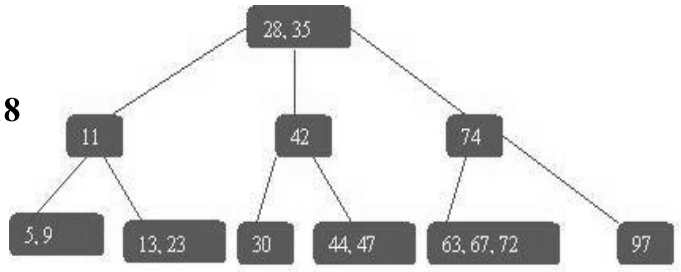
Thêm vào

- Các mục dữ liệu mới luôn luôn được chèn vào tại các node lá.
- Việc thêm vào cây 2-3-4 trong bất cứ trường hợp nào thì quá trình cũng bắt đầu bằng cách tìm kiếm node lá phù hợp.

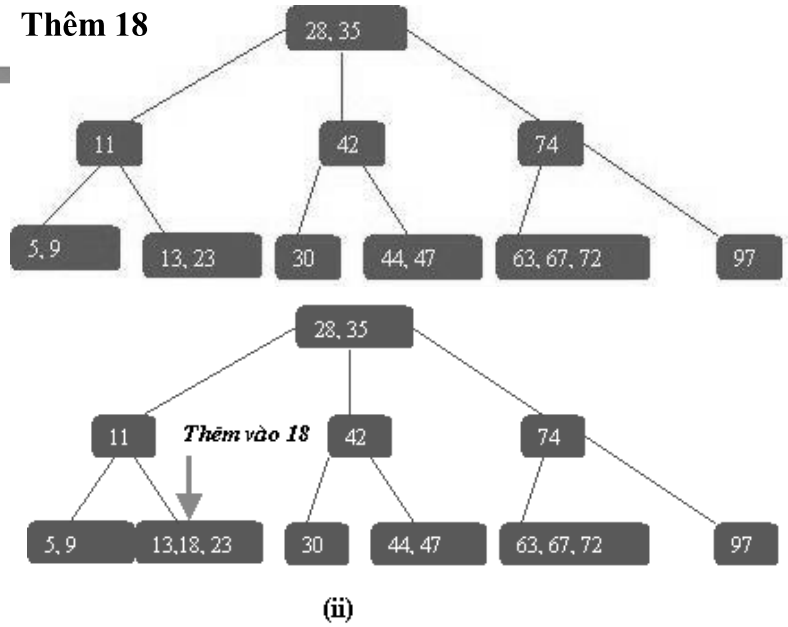
Thêm vào

- Nếu không có nút đầy nào (nút có đủ 3 mục dữ liệu) được tìm thấy trong quá trình tìm kiếm, việc chèn vào khá là dễ dàng.
- Khi nút lá phù hợp được tìm thấy, mục dữ liệu mới đơn giản là thêm vào nó

Thêm 18



Thêm 18



(ii)

Thêm vào (tt)

- Việc thêm vào sẽ trở nên phức tạp hơn nếu gặp phải một node đầy (node có số mục dữ liệu đầy đủ) trên nhánh dẫn đến điểm thêm vào. Khi điều này xảy ra, node này cần thiết phải được tách ra.
- Quá trình tách nhằm giữ cho cây cân bằng.
- Loại cây 2-3-4 mà chúng ta đề cập ở đây thường được gọi là cây 2-3-4 top-down (các node được tách ra theo hướng đi xuống điểm chèn)

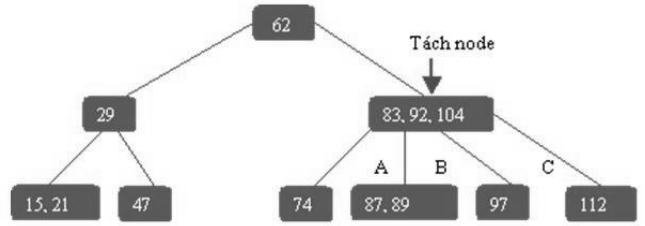
Thêm vào (tt)

- Giả sử ta đặt tên các mục dữ liệu trên node bị phân chia là **A**, **B** và **C**. Sau đây là tiến trình tách (chúng ta giả sử rằng node bị tách không phải là node gốc; chúng ta sẽ kiểm tra việc tách node gốc sau này)
  - Một node mới và rỗng được tạo. Nó là anh em với node sẽ được tách và được đưa vào bên phải của nó.
  - Mục dữ liệu C được chuyển vào node mới.
  - Mục dữ liệu B được chuyển vào node cha của node được tách.
  - Mục dữ liệu A không thay đổi.
  - Hai node con bên phải nhất bị hủy kết nối từ node được tách và kết nối đến node mới.

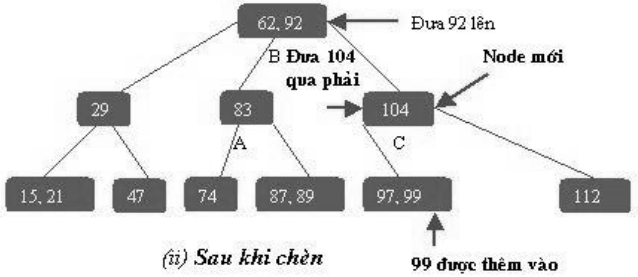
Thêm vào (tt)

Thêm 99

a) Trước khi thêm 99



b) Sau khi thêm 99



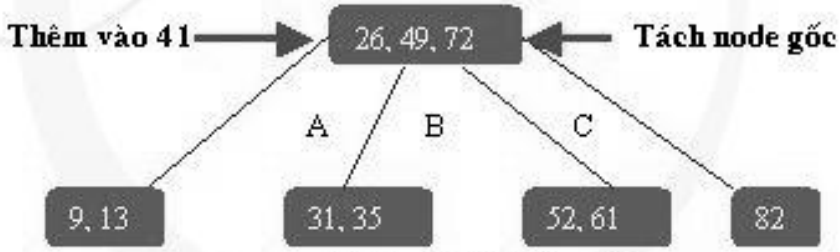
Thêm vào (tt)

◆ **Tách node gốc**

Khi gặp phải node gốc đầy tại thời điểm bắt đầu tìm kiếm điểm chèn, kết quả của việc tách thực hiện như sau:

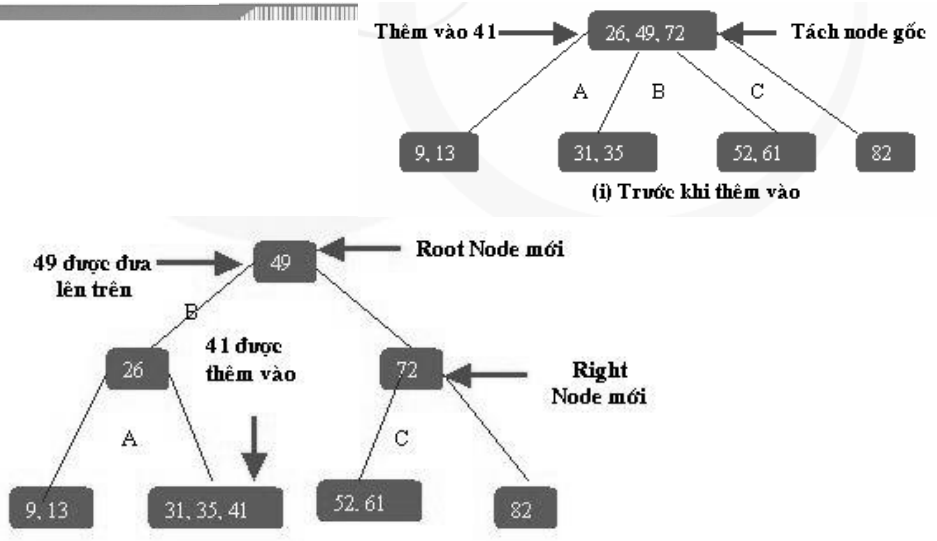
- Node mới được tạo ra để trở thành gốc mới và là cha của node được tách.
- Node mới thứ hai được tạo ra để trở thành anh em với node được tách.
- Mục dữ liệu C được dịch chuyển sang node anh em mới.
- Mục dữ liệu B được dịch chuyển sang node gốc mới.
- Mục dữ liệu A vẫn không đổi.
- Hai node con bên phải nhất của node được phân chia bị hủy kết nối khỏi nó và kết nối đến node mới bên phải.

Thêm vào (tt)



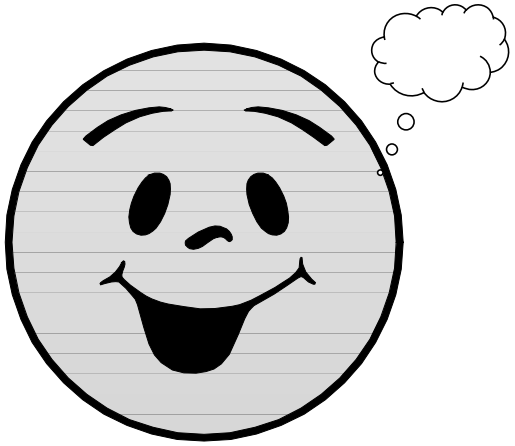
(i) Trước khi thêm vào

Thêm vào (tt)



(i) Trước khi thêm vào

(ii) Sau khi thêm vào



Cảm ơn !