



B-trees

Lịch sử

- Được đưa ra bởi Rudolf Bayer và Edward M McCreight năm 1972
- Mục đích
 - Đưa ra một CTDL cây cho phép tìm kiếm, truy cập, chèn, xóa trong thời gian logarithm.
 - Được tối ưu cho các hệ thống đòi hỏi phải đọc và ghi các khối dữ liệu lớn

Vấn đề

- Cần lưu trữ số phần tử dữ liệu rất lớn
- Lưu trữ trên bộ nhớ ngoài
- Tìm kiếm nhanh

⇒ Sử dụng một loại cây khác – B Tree

- Cây có nhiều nhánh => giảm chiều cao của cây
- Gom dữ liệu thành các block => giảm số lần truy cập đĩa

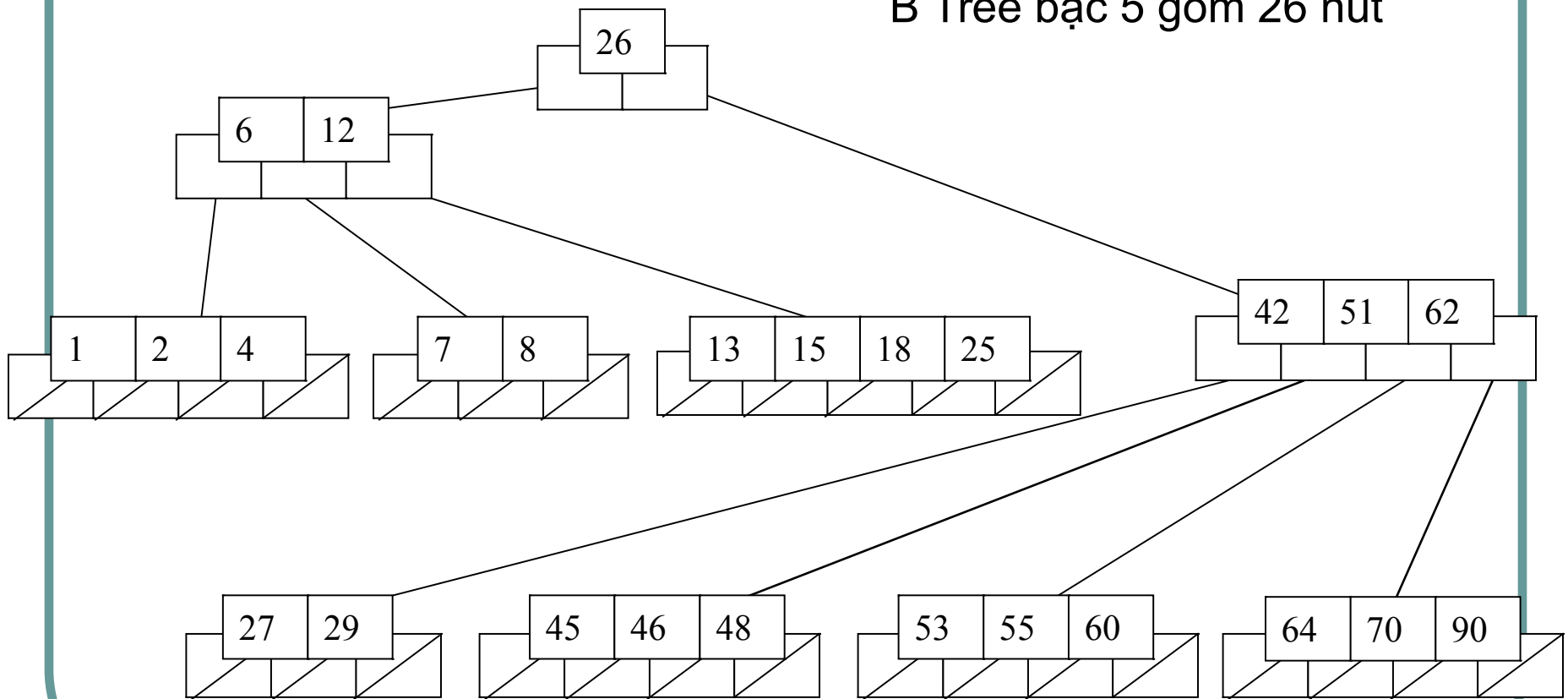
Định nghĩa

Một B-Tree bậc m có các tính chất (1972)

- Các nút lá có cùng mức
- Số lượng khóa của các nút khác lá = số lượng con - 1
- Các nút (trừ nút gốc, nút lá) có ít nhất $\lceil m / 2 \rceil$ con
- Nút gốc hoặc là nút lá hoặc có từ 2 đến m con
- Nút lá bao gồm nhiều hơn $m-1$ khóa
- m là số lẻ

Ví dụ về B-Tree

B Tree bậc 5 gồm 26 nút



Các ứng dụng của B Tree

- Cây đồ-đen: được sử dụng rộng rãi trong các bảng kí hiệu hệ thống
 - Java: `java.util.TreeMap`, `java.util.TreeSet`.
 - C++ STL: `map`, `multimap`, `multiset`.
 - Linux kernel: `linux/rbtree.h`.
- B-Trees: được sử dụng rộng rãi trong hệ quản trị CSDL và hệ thống file
 - Windows: HPFS.
 - Mac: HFS, HFS+.
 - Linux: ReiserFS, XFS, Ext3FS, JFS.
 - Databases: ORACLE, DB2, INGRES, SQL, PostgreSQL
- Các nút trong B-Tree được lưu trữ trong bộ nhớ ngoài (ổ đĩa), thay vì bộ nhớ trong (memory)
- Các thao tác truy cập: *Disk-Read()*, *Disk-Write()*, *Allocate-Node()*

B-Tree Library

- Link:
- <http://www.hydrus.org.uk/doc/bt/html/index.htm>
|

API

- Tạo một file BTree

BTA* btcrt(char* *fid*, int *nkeys*, int *shared*);

- Mở file B Tree

BTA* btopn(char* *fid*, int *mode*, int *shared*);

- Đóng file B Tree

int btcls(BTA* *btact*);

API (cont.)

- Chèn một khóa & giá trị

```
int btins(BTA* btact, char* key, char* data, int dsize);
```

- Cập nhật giá trị cho khóa đã có

```
int btupd(BTA* btact, char* key, char* data, int dsize);
```

- Xác định giá trị cho khóa đã có

```
int btisel(BTA* btact, char* key, char* data, int dsize, int*  
rsize);
```

- Xóa một khóa & giá trị của nó

```
int btdel(BTA* btact, char* key);
```

- Xác định giá trị cho khóa tiếp theo

```
int btseIn(BTA* btact, char* key, char* data, int dsize, int*  
rsize);
```

Xây dựng & cài đặt thư viện B Tree

- Giải nén vào một thư mục nào đó.

```
$cd <bt library>
```

```
$make clean
```

```
$make
```

- Tạo ra file thư viện libbt.a, chương trình test (harness) bt, và tiện ích đánh chỉ mục kcp

Bài 1

- Cài đặt & biên dịch thư viện BT
- Chạy harnessbt để xác định đã cài đặt & biên dịch thành công
- Tài liệu liên quan:

<http://www.hydrus.org.uk/doc/bt/html/ch05.htm>

Bài 2

- Sử dụng thư viện BT để viết chương trình quản lý danh bạ điện thoại, lưu dữ liệu ở bộ nhớ ngoài
- Chức năng
 - Thêm một thông tin liên lạc mới
 - Tìm kiếm theo tên
 - Hiển thị toàn bộ danh bạ
 - Xóa một thông tin liên lạc

Thư viện khác về B-Tree

- Download tại

<http://www.mycplus.com/utilitiesdetail.asp?iPro=10>

- Thư viện cho phép xác định các hàm so sánh khác nhau cho khóa

Project 1

- Chương trình quản lý từ điển máy tính. Có chức năng
 - Add/Search/Delete từ (sử dụng B-Tree)
 - Hỗ trợ hoàn thiện tìm kiếm. Ví dụ khi gõ “comput” và ấn <tab>, từ “computer” sẽ được hoàn thiện (giống trong Bash Shell)
 - Gợi ý search => sử dụng soundex library
- Test chương trình với từ điển có hàng triệu từ
- Nhóm có 3-4 người
- Cần có giao diện đồ họa
- Trong chương trình chỉ rõ: tham khảo mã nguồn khóa trước hay không, chỉ ra tài liệu tham khảo và nội dung tham khảo

Yêu cầu tiếp

- Nhóm hoàn thiện tất cả các bài tập trong slide từ tuần 1 cho đến tuần bảo vệ Project (dự kiến tuần 10)
- Đánh giá dựa trên:
 - Báo cáo (gồm Project + Bài tập) + chương trình + Trình bày
 - Gian dối trong Project: cả nhóm 0 điểm
 - Thành viên trong nhóm không nắm rõ về bài tập trong slide: trừ 1đ cả nhóm