

balance tree

B-trees

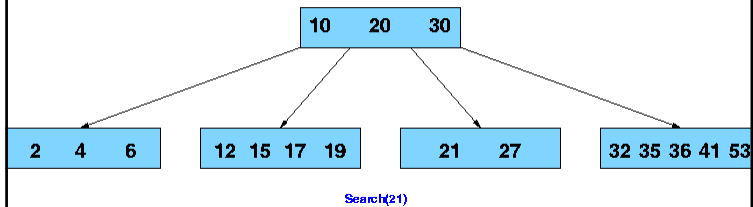
Created by anhtt-fit@mail.hut.edu.vn
Edited by huonglt-fit@mail.hut.edu.vn

B-Tree

- Sinh cây 2-3-4 bằng cách cho phép tối đa M liên kết mỗi nút.
- Ứng dụng chính: quản lý các file hệ thống.
 - Đọc 1 trang từ đĩa vào bộ nhớ khá tốn kém.
 - Truy cập thông tin trong 1 trang bộ nhớ không tốn thời gian.
 - Mục đích: tối thiểu hoá số trang truy cập
 - Kích thước 1 nút = kích thước trang
- Cân bằng giữa không gian – thời gian
 - M lớn: chỉ có 1 số ít mức trên cây (độ cao của cây nhỏ)
 - M nhỏ: ít tốn không gian
 - Số trang truy cập = $\log_M N$ / phép toán
 - Với $M = 1000$, $N < 1$ trillion.

Tìm kiếm

B-Tree: Minimization Factor $t = 3$, Minimum Degree = 2, Maximum Degree = 5



Chèn

Các ứng dụng của B-Tree

- Red-black trees: được dùng rộng rãi cho bảng ký hiệu hệ thống
 - Java: java.util.TreeMap, java.util.TreeSet.
 - C++ STL: map, multimap, multiset.
 - Linux kernel: linux/rbtree.h.
- B-Trees: được dùng rộng rãi cho các hệ thống file và CSDL
 - Windows: HPFS.
 - Mac: HFS, HFS+.
 - Linux: ReiserFS, XFS, Ext3FS, JFS.
 - Databases: ORACLE, DB2, INGRES, SQL, PostgreSQL
- Giả thiết tất cả các nút trong B-Tree được lưu trong đĩa, không lưu trong bộ nhớ
- Các thao tác cơ bản để truy cập trang: *Disk-Read()*, *Disk-Write()*, *Allocate-Node()*

B-Tree Library

- Lấy phần mềm và tài liệu tại <http://www.hydrus.org.uk/doc/bt/html/index.html>

API

- Tạo B Tree File *khởi tạo = 0; b tree tự động tăng*
`BTA* btcrct(char* fid, int nkeys, int shared);`
- Mở B Tree File *=0 nếu không cho phép share*
`BTA* btopen(char* fid, int mode, int shared);`
- Đóng B Tree File
`int btcls(BTA* btact);`

fid=file save

API (tiếp)

- Chèn 1 khóa và dữ liệu *kích thước của data*
`int btins(BTA* btact, char* key, char* data, int dsize);`
- Cập nhật dữ liệu cho khóa có sẵn
`int btupd(BTA* btact, char* key, char* data, int dsize);`
- Xác định dữ liệu cho khóa có sẵn *tim sdt*
`int btisel(BTA* btact, char* key, char* data, int dsize, int* rsize);` *int=0 nếu thành công*
- Xóa khóa và dữ liệu tương ứng
`int btldel(BTA* btact, char* key);`
- Xác định dữ liệu cho khóa tiếp theo
`int btsele(BTA* btact, char* key, char* data, int dsize, int* rsize);` *rsizesẽ đc trả về.*

`btinit();`



Xây dựng và cài đặt thư viện BT

- Giải nén file tar vào 1 thư mục nào đó.
\$cd <bt library>
\$make clean
\$make
- Tạo ra thư viện **libbt.a** (câu lệnh make), một chương trình thử nghiệm BT **bt**, và 1 chương trình **kcp** cho phép sao chép các file chỉ số BT.

Quiz 1

- Cài đặt và dịch thư viện BT vào máy
- Chạy BT test harness để kiểm tra xem thư viện BT đã được cài đặt thành công chưa
- Tham khảo tài liệu tại
<http://www.hydrus.org.uk/doc/bt/html/ch05.htm>

html

Quiz 2

- Sử dụng thư viện BT để viết chương trình phonebook để thao tác với dữ liệu trên đĩa.

Thư viện B-Tree khác

- Tải về tại
<http://www.mycplus.com/utilitiesdetail.asp?iPro=10>
- Thư viện này cho phép xác định các hàm so sánh khác nhau cho các khóa

Bài tập lớn 1

- Xây dựng 1 chương trình để quản lý từ điển điện tử
 - Thêm/Tìm kiếm/Xóa 1 từ (dùng B-Tree)
 - Tìm kiếm kiểu “auto complete”. Ví dụ, khi gõ “comput” và <tab>, từ “computer” cần được hiện lên tự động (giống như trong Bash Shell)
 - Tìm kiếm có gợi ý => Sử dụng thư viện soundex
- Hãy thử nghiệm hiệu năng của chương trình của bạn với 1 từ điển gồm vài triệu từ (các từ có thể tạo ngẫu nhiên)
 - Thử nghiệm với 2 thao tác cơ bản: tìm kiếm và chèn
- Làm việc theo nhóm 3-4 sinh viên