



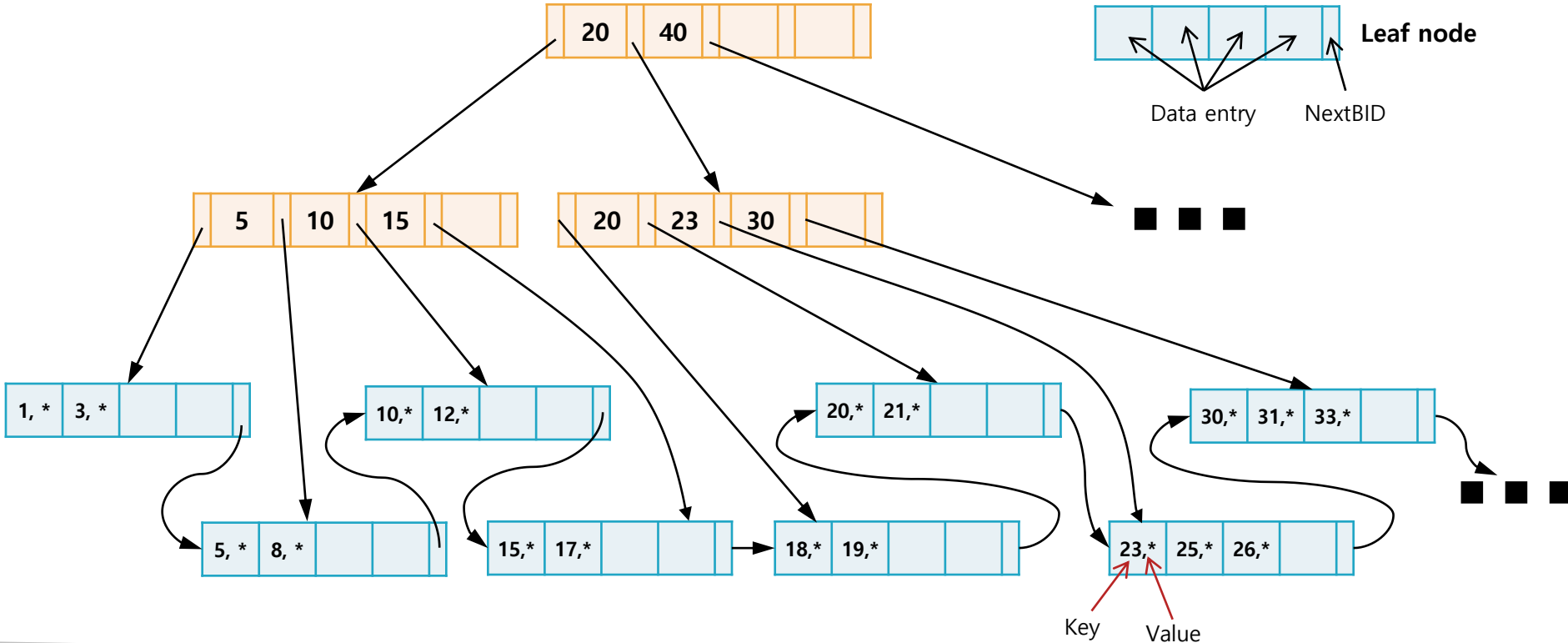
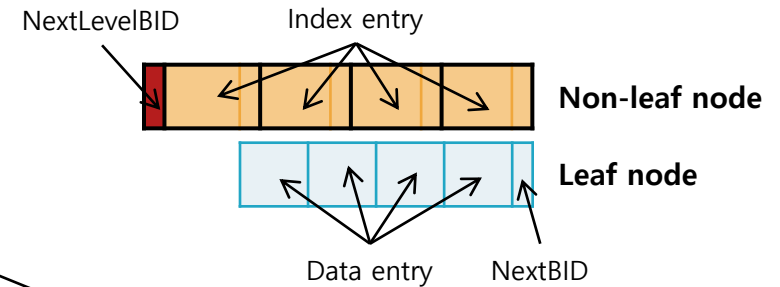
Implementation of a Disk Based B⁺-Tree

CSE3207 Database Project #2

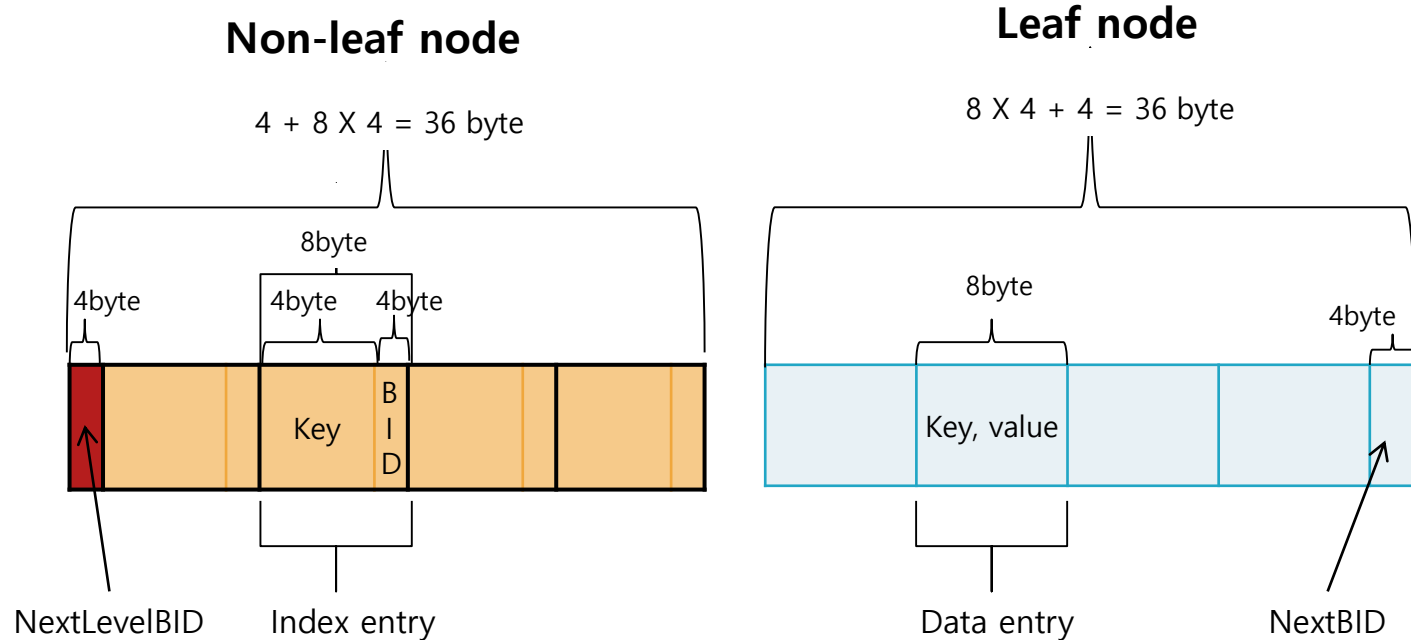
Assignment Date : May 16th, 2019
Due Date : June 21th, 2019



B⁺-Tree Structure

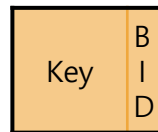


Details of Nodes and Entries

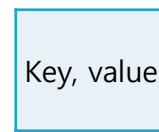


* page size = node size = 36 byte

Index entry
Key
NextLevelBID



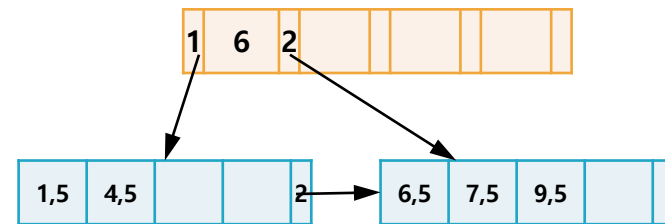
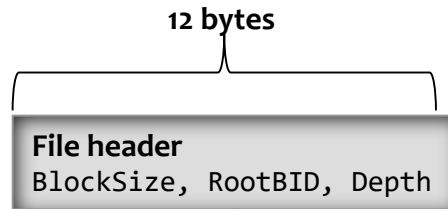
Data entry
Key
Value



○○○ B⁺-Tree Data File Structure ○○○

Insert data (key, ID)

1	5
6	5
4	5
7	5
9	5



*36 bytes for each node

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f	
00000000h:	24	00	00	00	03	00	00	00	01	00	00	00	01	00	00	00	;
00000010h:	05	00	00	00	04	00	00	00	05	00	00	00	00	00	00	00	;
00000020h:	00	00	00	00	00	00	00	00	00	00	00	00	02	00	00	00	;
00000030h:	06	00	00	00	05	00	00	00	07	00	00	00	05	00	00	00	;
00000040h:	09	00	00	00	05	00	00	00	00	00	00	00	00	00	00	00	;
00000050h:	00	00	00	00	01	00	00	00	06	00	00	00	02	00	00	00	;
00000060h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	;
00000070h:	00	00	00	00	00	00	00	00									;


84 bytes from the file starting position

Physical offset of a PageID = 12 + ((BID-1) * BlockSize)



Test & UI



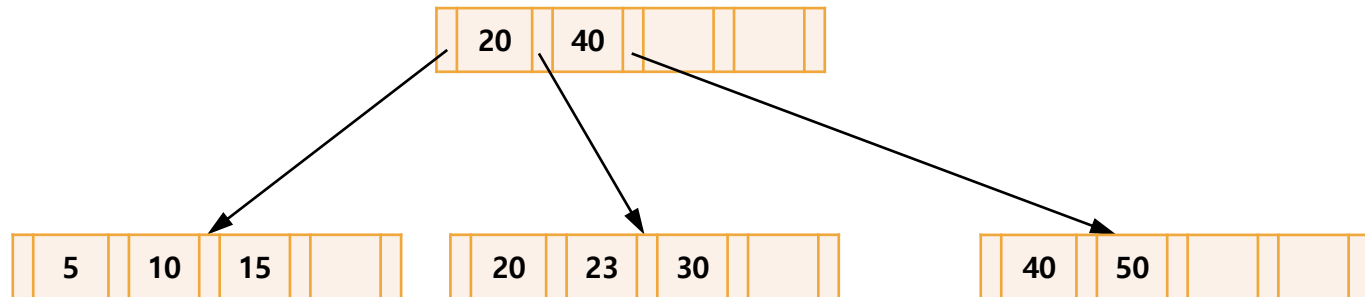
- ▶ Index creation
 - ▶ *btree.exe c [btree binary file] [page_size], e.g., [btree.exe c btree.bin 36](#)*
 - ▶ Generates [btree binary file] with only header
- ▶ Insertion
 - ▶ *btree.exe i [btree binary file] [records text file], e.g., [btree.exe i btree.bin insert.txt](#)*
 - ▶ Inserts nodes(entries) to [btree binary file] using [records text file]
- ▶ Point(exact) search
 - ▶ *btree.exe s [btree binary file] [input text file] [output  file],*
e.g., [btree.exe i btree.bin search.txt output.txt](#)
 - ▶ Output searched keys and IDs to [output text file] using [btree binary file]
- ▶ Range search
 - ▶ *btree.exe r [btree binary file] [input text file] [output text file],*
e.g., [btree.exe r btree.bin rangearch.txt output.txt](#)
 - ▶ Output searched keys and IDs to [output text file] using [btree binary file]
- ▶ **MUST** follow input and output file **formats** in the document



Test & UI



- ▶ Print B⁺-Tree structure
 - ▶ *btree.exe p [btree binary file] [output text file]*
 - ▶ Output node structure of [btree binary file] to [output text file]
 - ▶ Output only root node<level 0> and next level <level1>
 - ▶ Example



```
<0>
20, 40
<1>
5, 10, 15, 20, 23, 30, 40, 50
```



Submission



- ▶ To the I-Class website
- ▶ Upload a zip file containing the followings:
 - ▶ A single source file, named as “***btree.cpp or btree.c***”
 - ▶ README.doc explaining:
 - ▶ What you’ve implemented and what you’ve NOT
 - ▶ Brief explanation of your implementation (Do not make it look fancy, less than 0.5 page)
 - ▶ How to compile and run
 - ▶ Talk about your experience of doing this project
 - ▶ Contact information (just in case)

