

# Indexes

By aak1247

# CONTENTS

◀ Hash

◀ Tree

1

Hash

Column Value ->Key->Index

最常见：倒排索引、部分索引

单词ID	单词	文档频率	倒排列表 ( DocID;TF;<POS> )
1	谷歌	5	(1;1;<1>), (2;1;<1>), (3;2;<1;6>), (4;1;<1>), (5;1;<1>)
2	地图	5	(1;1;<2>), (2;1;<2>), (3;1;<2>), (4;1;<2>), (5;1;<2>)
3	之父	4	<1;1;<3>), (2;1;<3>), (4;1;<3>), (5;1;<3>)
4	跳槽	2	(1;1;<4>), (4;1;<4>)
5	Facebook	5	(1;1;<5>), (2;1;<5>), (3;1;<8>), (4;1;<5>), (5;1;<8>)
6	加盟	3	(2;1;<4>), (3;1;<7>), (5;1;<5>)
7	创始人	1	(3;1;<3>)
8	拉斯	2	(3;1;<4>), (5;1;<4>)
9	离开	1	(3;1;<5>)
10	与	1	( 4;1;<6>)

Nebula Graph

Key				Value
PartitionId	VertexId	TagId	Version	Property values

Key				Value
PartitionId	IndexId	Index binary	VertexId	

Key						Value
PartitionId	SrcVertexId	EdgeType	EdgeRank	DstVertexId	Version	Prop values

Key						Value
PartitionId	IndexId	Index binary	SrcVertexId	EdgeRank	DstVertexId	

# Hash – 在NoSQL/NewSQL中的应用

Tidb

Key: tablePrefix{TableID}\_recordPrefixSep{RowID}

Value: [col1, col2, col3, col4]

Key: tablePrefix{tableID}\_indexPrefixSep{indexID}\_indexedColumnsValue

Value: RowID

HBase

Row Key	CF	Column	TimeStam p	Value
Row 1	info	name	1546679322 575	张三

知乎 @spacedong

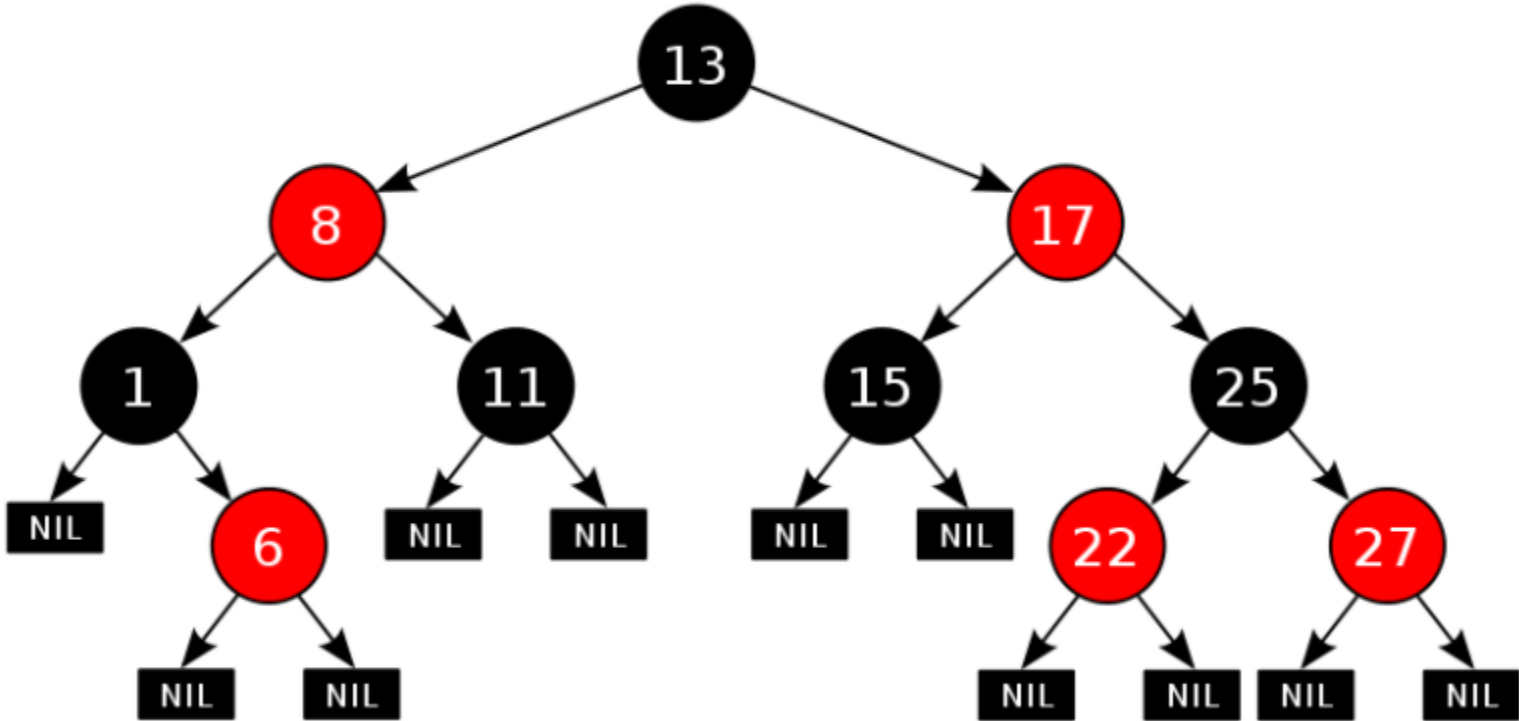


2

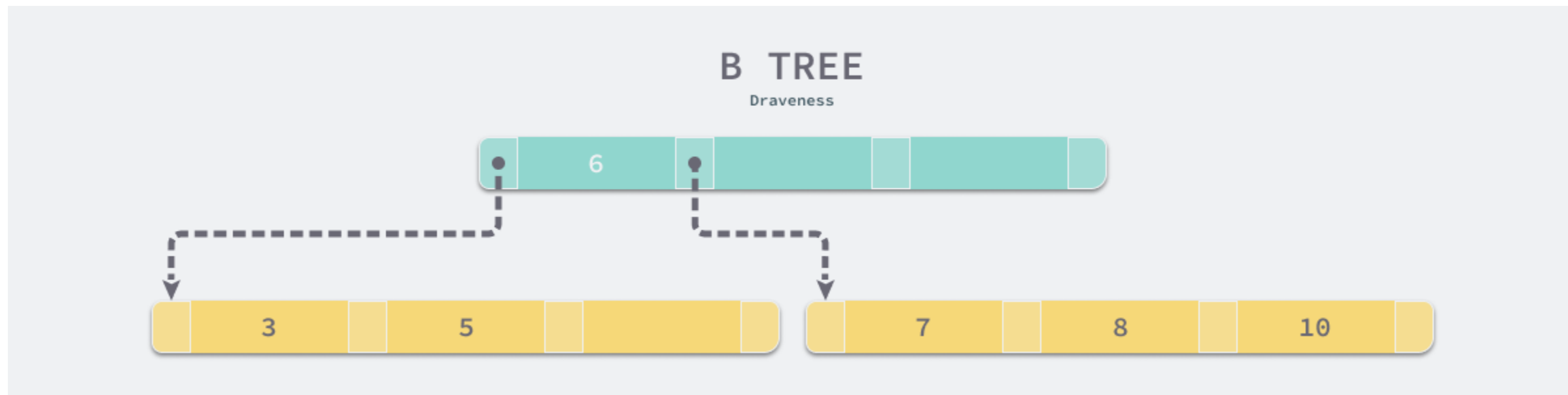
Tree

红黑树、AVL树、跳表  
B树、B+树、LSM树

- 1.每个结点要么是红的要么是黑的。
- 2.根结点是黑的。
- 3.每个叶结点（叶结点即指树尾端NIL指针或NULL结点）都是黑的。
- 4.如果一个结点是红的，那么它的两个儿子都是黑的。
- 5.对于任意结点而言，其到叶结点树尾端NIL指针的每条路径都包含相同数目的黑结点。

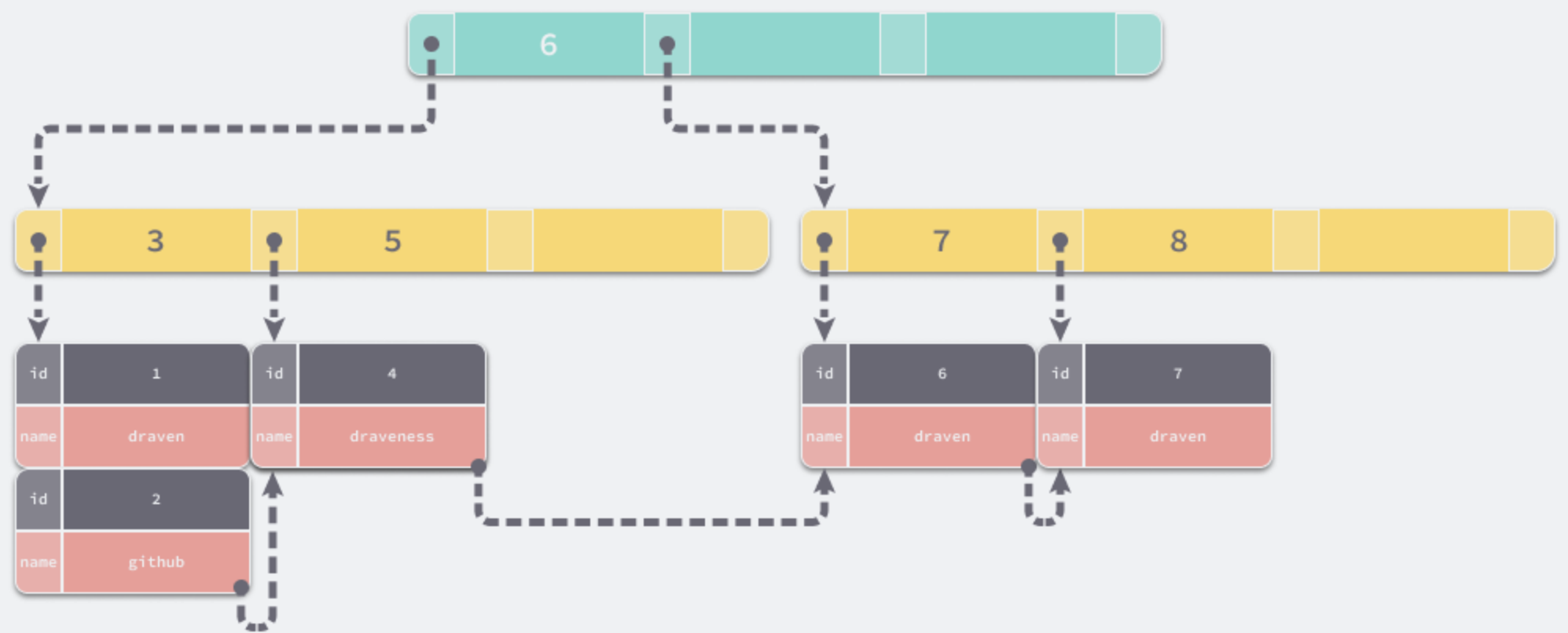


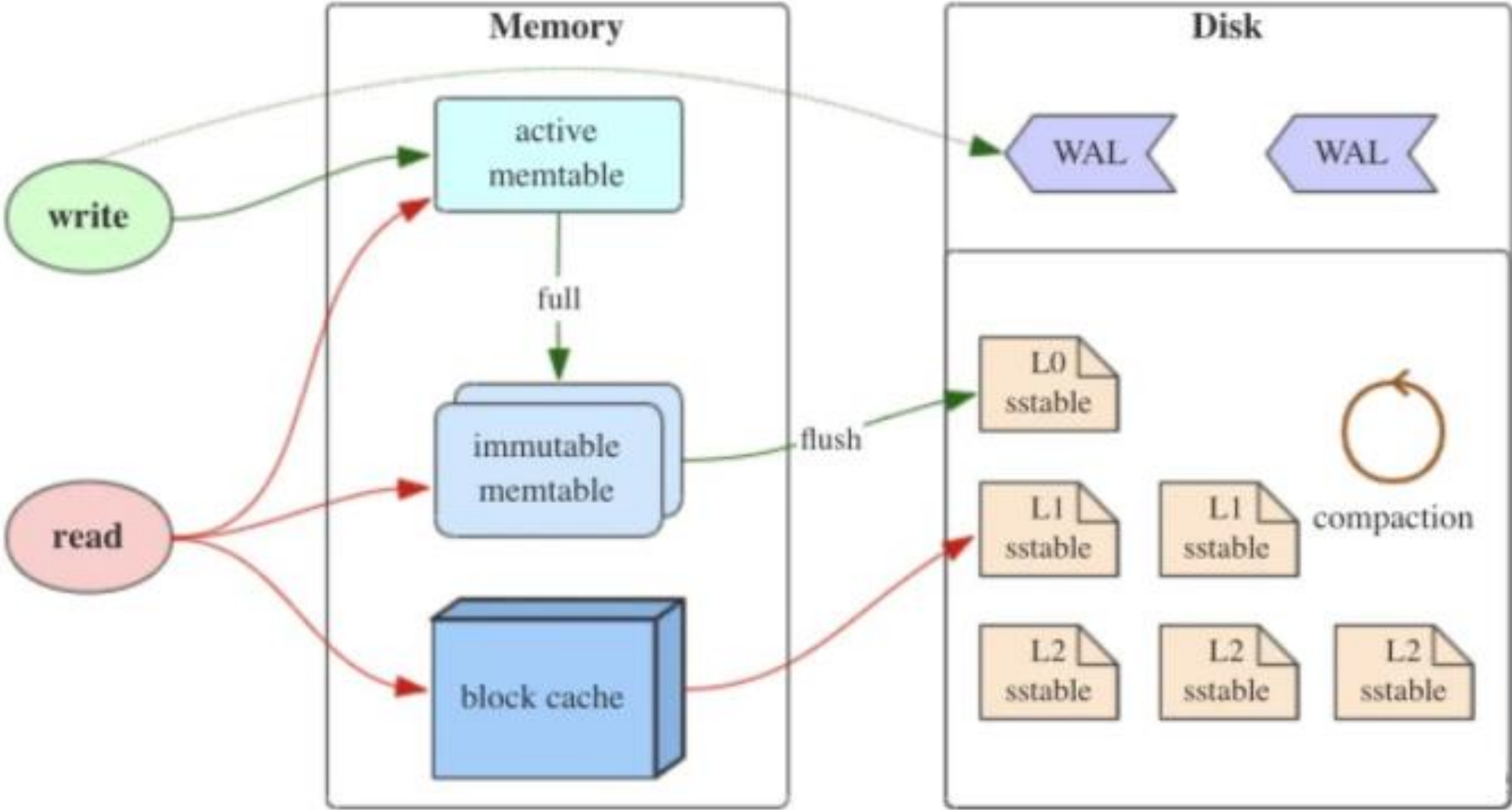
- (1) 排序方式：所有节点关键字是按递增次序排列，并遵循左小右大原则；
- (2) 子节点数：非叶节点的子节点数 $>1$ ，且 $\leq M$ ，且 $M \geq 2$ ，空树除外（注：M阶代表一个树节点最多有多少个查找路径， $M=M$ 路,当 $M=2$ 则是2叉树, $M=3$ 则是3叉）；
- (3) 关键字数：枝节点的关键字数大于等于 $\text{ceil}(m/2)-1$ 个且小于等于 $M-1$ 个（注： $\text{ceil}()$ 是个朝正无穷方向取整的函数 如 $\text{ceil}(1.1)$ 结果为2）；
- (4) 所有叶子节点均在同一层、叶子节点除了包含了关键字和关键字记录的指针外也有指向其子节点的指针只不过其指针地址都为null对应下图最后一层节点的空格子；



MYSQL INNODB B+ TREE

Draveness







Left-Leaning Red-Black Trees, Dagstuhl Workshop on Data Structures, Wadern, Germany, February, 2008

<https://www.cs.umb.edu/~poneil/lsmtree.pdf> LSM tree

Thank You!