2-11 Heapsort

Hengfeng Wei

hfwei@nju.edu.cn

June 02, 2020

Organization and Maintenance of Large Ordered Indexes

R. Bayer and E. McCreight

Received September 29, 1971

Summary. Organization and maintenance of an index for a dynamic random access file is considered. It is assumed that the index must be kept on some pseudo random access backup store like a disc or a drum. The index organization described allows retrieval, insertion, and deletion of keys in time proportional to $\log_k I$ where I is the size of the index and k is a device dependent natural number such that the performance of the scheme becomes near optimal. Storage utilization is at least 50% but generally much higher. The pages of the index are organized in a special data-structure, so-called B-trees. The scheme is analyzed, performance bounds are obtained, and a near optimal k is computed. Experiments have been performed with indexes up to 100000 keys. An index of size 15000 (100000) can be maintained with an average of 9 (at least 4) transactions per second on an IBM 360/44 with a 2311 disc.

Organization and Maintenance of Large Ordered Indexes

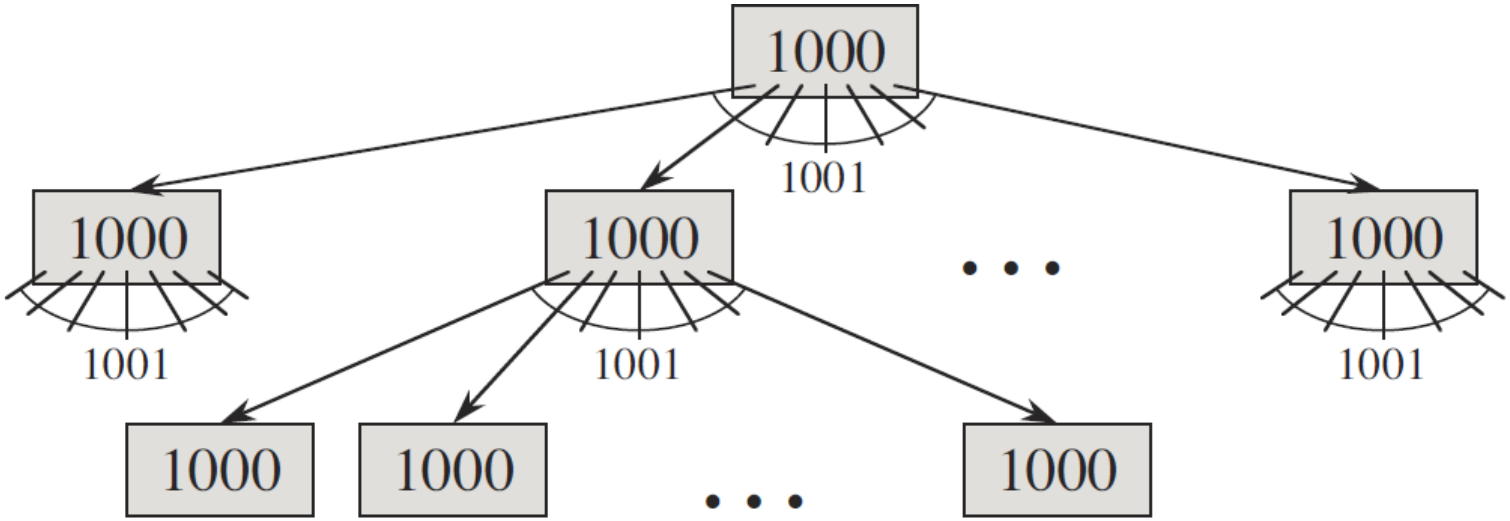
R. Bayer and E. McCreight

Received September 29, 1971

Summary. Organization and maintenance of an index for a dynamic random access file is considered. It is assumed that the index must be kept on some pseudo random access backup store like a disc or a drum. The index organization described allows retrieval, insertion, and deletion of keys in time proportional to $\log_k I$ where I is the size of the index and k is a device dependent natural number such that the performance of the scheme becomes near optimal. Storage utilization is at least 50% but generally much higher. The pages of the index are organized in a special data-structure, so-called B-trees. The scheme is analyzed, performance bounds are obtained, and a near optimal k is computed. Experiments have been performed with indexes up to 100000 keys. An index of size 15000 (100000) can be maintained with an average of 9 (at least 4) transactions per second on an IBM 360/44 with a 2311 disc.

"Bayer and McCreight introduced B-trees in 1972;

they did not explain their choice of name."



2-way *vs.* multi-way

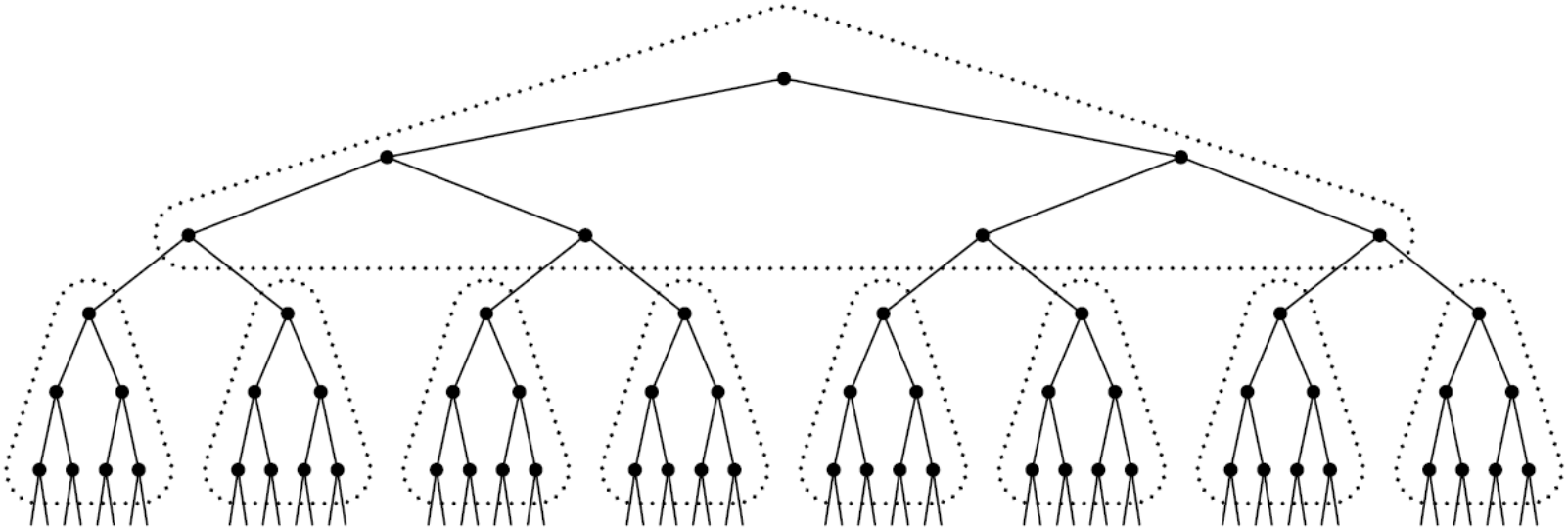
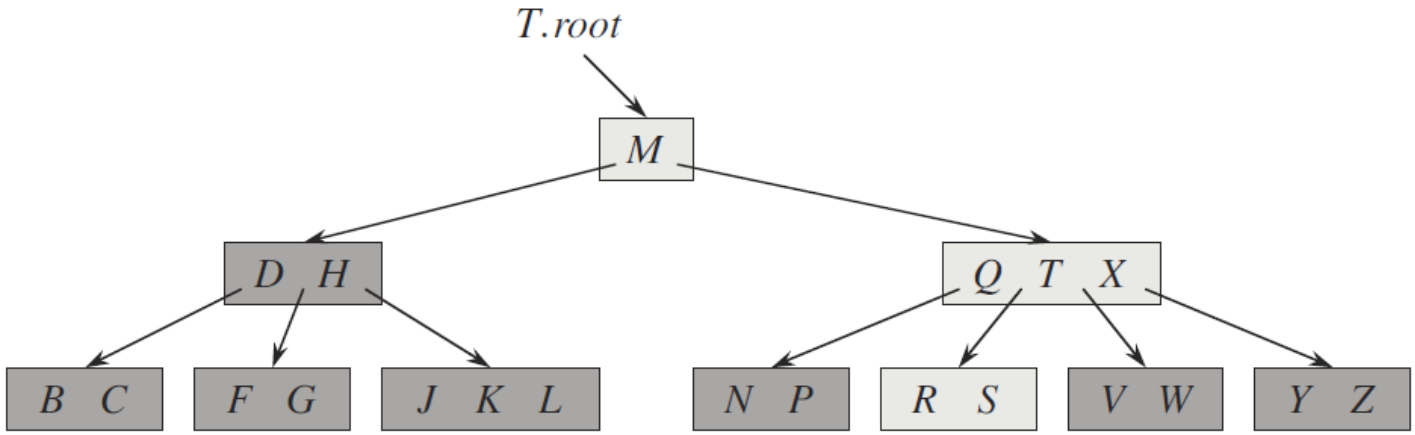


Fig. 29. A large binary search tree can be divided into "pages."

node *vs.* pages

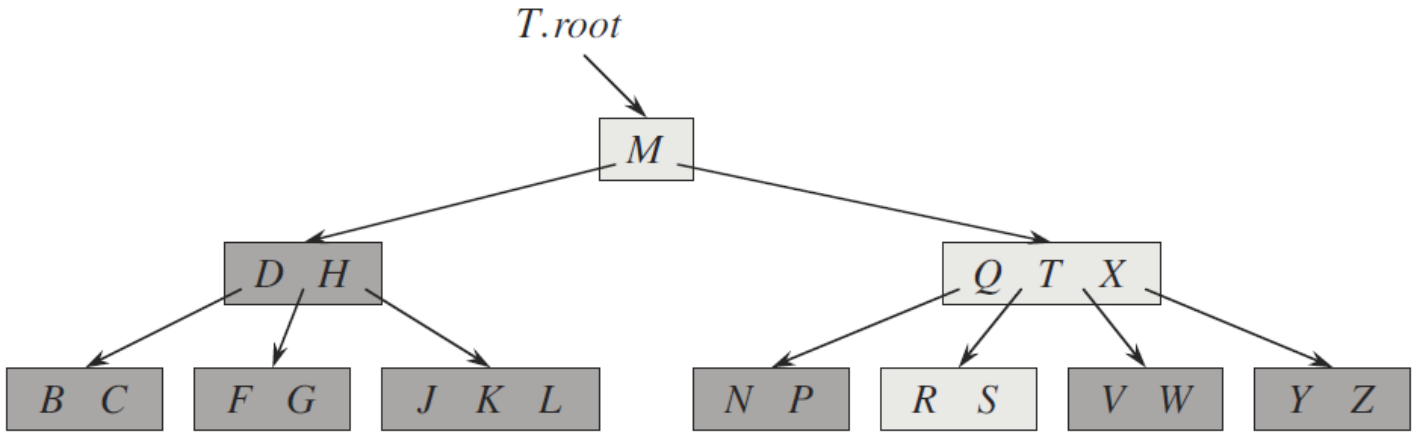
Minimum (TC 18.2-3)

Explain how to find the minimum key stored in a B-tree.



Minimum (TC 18.2-3)

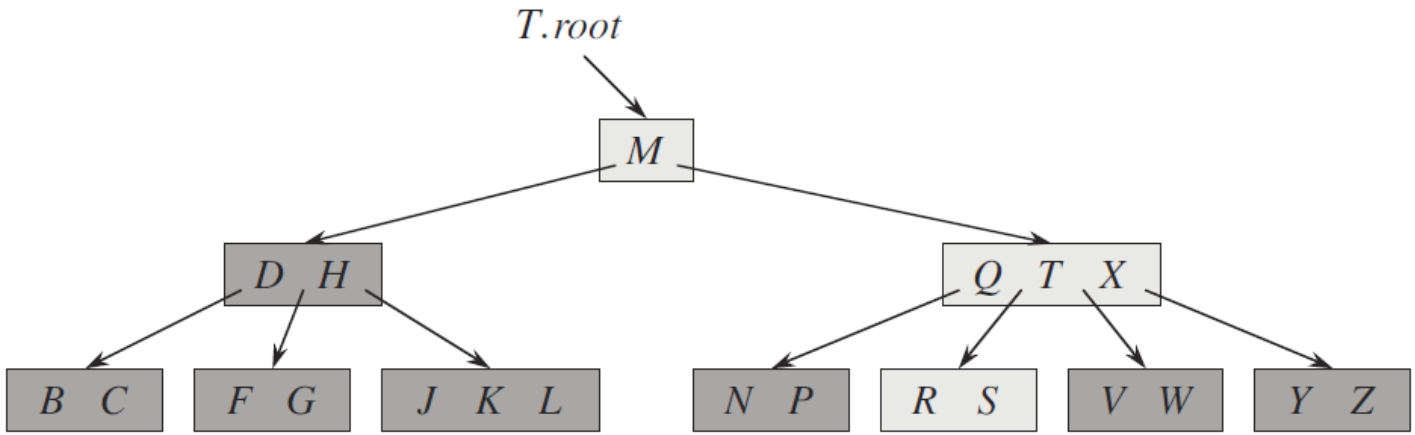
Explain how to find the minimum key stored in a B-tree.



the leftmost key in the leftmost node

Predecessor (TC 18.2-3)

Explain how to find the predecessor of a given key stored in a B-tree.



Insertion (TC 18.2-4*)

Suppose that we insert the keys $\{1, 2, \ldots, n\}$ in increasing order into an empty B-tree with minimum degree 2.

How many nodes, denoted $T(n)$, does the final B-tree have?

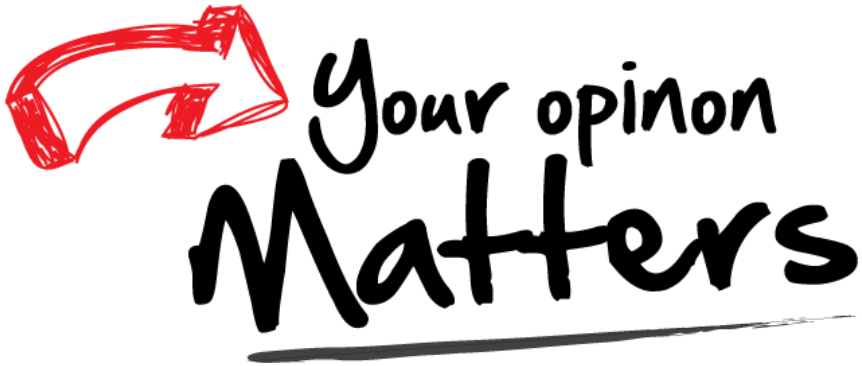
Insertion (TC 18.2-4*)

Suppose that we insert the keys $\{1, 2, \ldots, n\}$ in increasing order into an empty B-tree with minimum degree 2.

How many nodes, denoted $T(n)$, does the final B-tree have?

$$T_0 = 1$$


Thank
You!



Office 302

Mailbox: H016

hfwei@nju.edu.cn