数理解析研究所講究録
第423巻 1981年 116-127

重み付きレコードを扱うB^*木のページネーションの改良について

九州大学理学部　　有川節夫

McCreight [1] による可変長レコードを扱うB^*木を，少し拡張して重みつきレコードを扱うB^*木として把え，ページネーションのアルゴリズムをこれまでに知られていた$O(n \log n)$時間のものから$O(n)$時間のものへ改良したので，それについて述べる．同時に文献[1]におけるB^*木の定義について検討を行う．

1. B木

まず本稿で考えるB木について説明しよう．Kを全順序($<$)集合とする．Kの要素をキーと呼ぶ．キーxには関連情報α（番地や重みなど）が対応している．その組$r=(x, \alpha)$をレコードといい，レコードの有限集合をファイルという．（ディスク上の）番地付けのされた一定長の連続した領域をページという．以下では下図のような構造をしたページを考

p_0	(x_1, α_1)	p_1	(x_2, α_2)	p_2	//	$p_{\ell-1}$	(x_ℓ, α_ℓ)	p_ℓ	あき

えることにする. 図において, p_iは他のページへのポインタであり, (x_i, α_i)はレコードである.

<u>定義 1.</u> 上のような構造をしたページを節(node)としてもち, 次の条件を満たす有向木をクラス $\tau(m, h)$ のB木という: (mは偶数)

i) 根から各葉までのパスの長さはhである.

ii) 根以外の節は$\frac{1}{2}m \sim m$個のレコードをもつ.

iii) 葉でない節Pは P内のレコードの個数をℓとするとき, $\ell+1$個の子供をもつ.

iv) 節P内のレコードを $(x_1, \alpha_1), (x_2, \alpha_2), \cdots, (x_\ell, \alpha_\ell)$ とするとき

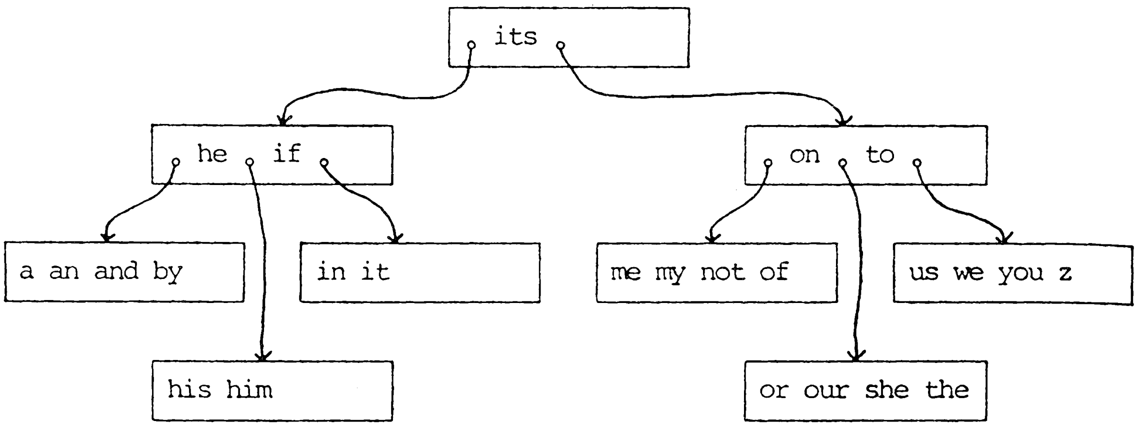
$$x_1 < x_2 < \cdots < x_\ell$$

である.

v) $P(p_i)$でポインタ p_i が指している節を表わし, $K(p_i)$ で $P(p_i)$ を根とする極大部分木の節内にあるキーの集合を表わす. このとき次が成立する:

$$\forall y \in K(p_0), \quad y < x_1$$
$$\forall y \in K(p_i), \quad x_i < y < x_{i+1} \quad (1 \leq i < \ell)$$
$$\forall y \in K(p_\ell), \quad x_\ell < y.$$

〈例〉 下図は$\tau(4,3)$のB木の例である．キーの集合は図にあるaからzまでの英単語であり，順序は辞書式である．



なお図において，ポインタは矢印で示し関連情報は省略した．

2. 重み付きレコードを扱うB^*木

レコードの挿入，削除の際にページをB木の条件を満足するように再編成する必要が生じる．本稿では，重いレコードがなるべくB木の根の近くにくるように再編成することを考える．

以下の議論に必要な約束を2，3しておこう．$p_0, p_1, \cdots, p_t$をポインタ，$r_1=(x_1, \alpha_1), \cdots, r_t=(x_t, \alpha_t)$を$x_i < x_{i+1}$ $(1 \le i < t)$なるレコードとする．このとき表現$S=p_0 r_1 p_1 r_2 p_2 \cdots p_{t-1} r_t p_t$をスクロール(scroll)という．また$t$をその長さといい$|S|$で示す．(B木の各ページPには長さ1～$m$

のスクロールが記録されていることになる．そこで，以下ではPでスクロールを表わすこともある．）レコード r とポインタ p の組（r, p）をエントリ（entry）と呼ぶ．レコード r_i の重みを $w(r_i)$ 又は単に $w(i)$ と書くことにする．また，B木の広がりを示す指数 m については $m=2n$ として話を進める．

レコードの挿入，削除はページの分割，連結，ページ内容の詰めかえを伴うが，これは根ページまで伝播することもある．以下で述べる操作は，こうした伝播の各段階における方略である．対象になるページをPとして，Pに対して／からエントリ（s, p_s）が挿入／削除されるものとする．

定義2. 以下の操作（I）（II）によって構成管理されるB木を B^* 木と呼ぶ．また（I）（II）の操作をページネーションと呼ぶ．

（I） レコードの挿入

Pが満員（full，$2n$ 個のレコードをもつ）でないときは，エントリ（s, p_s）をPの該当する場所に単に挿入するだけで済むから，以下ではPが満員の場合だけを考えることにする．

1） Pが根であるとき，Pにエントリ（s, p_s）を追加してできるスクロールを中央で分割して新しい根を作る．

2） Pのすぐ下の弟ページP′が満員でないとき；Pにエ

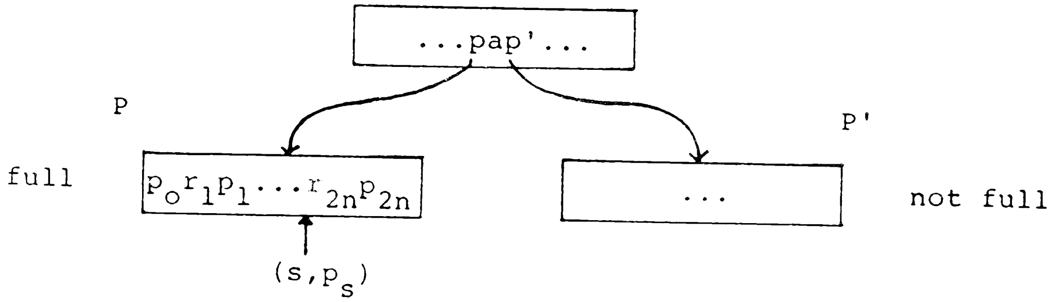
ントリ(s, ps)を追加してできるスクロール，親ページの境界レコード及びP′のスクロールを連結したスクロール

$$q_0 t_1 q_1 \cdots t_{2n+l+2}\ q_{2n+l+2} \quad (l=|p'|)$$

から，条件

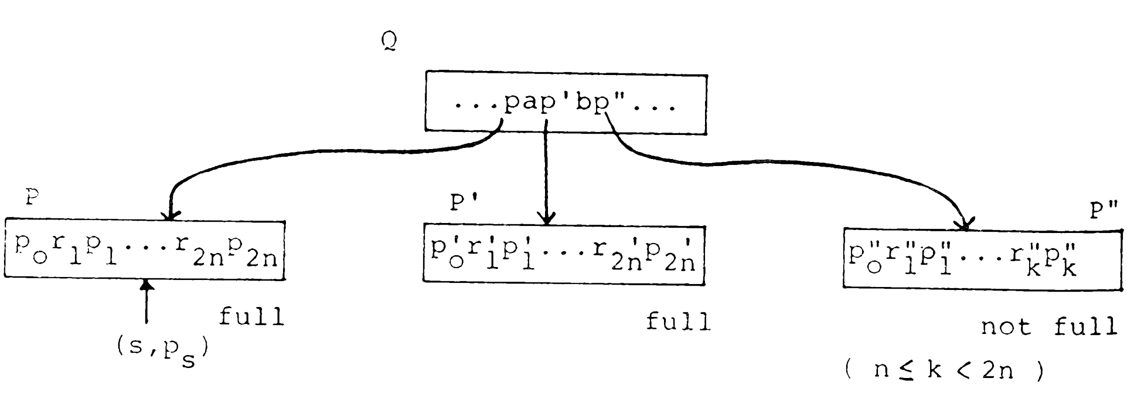
$$l+2 \le x \le n+l+2$$

の下で $w(t_x)$ を最大にする x を求め，$q_0 t_1 \cdots t_{x-1} q_{x-1}$ をPに残し，親ページの境界レコード q_{2n+2} $(=a)$ を t_x で置き換え，$q_{x+1} t_{x+2} \cdots q_{2n+l+2}$ をP′に置く.



なおPのすぐ上の兄ページP′が満員でないときも同様である.

3) Pのすぐ下の弟が満員で，その次の弟が満員でない（下図）のとき；



エントリ（S，p_s）とP，P′，P″のスクロール及びQの境界レコードa，bを綴り合せてできるスクロール

$$q_0 t_1 q_1 \cdots t_{4n+k+2}\, q_{4n+k+2}$$

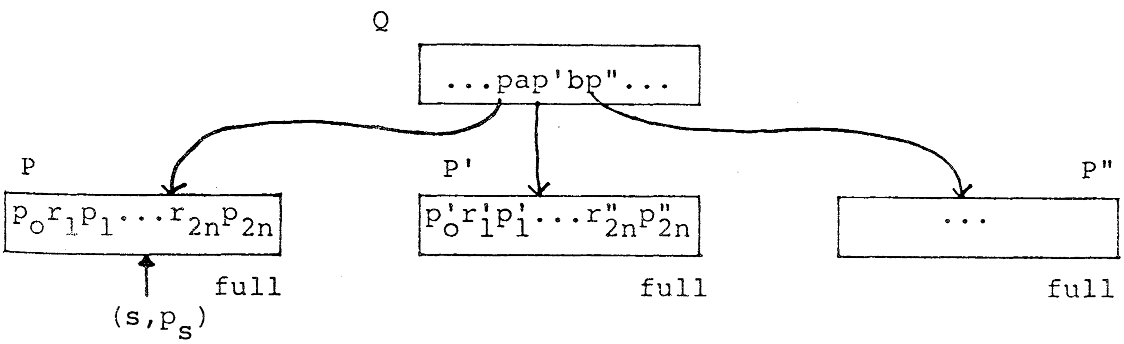
から，条件

$$\begin{cases} n+1 \le x \le 2n+1 \\ x+n+1 \le y \le x+2n+1 \\ 2n+k+2 \le y \le 3n+k+2 \end{cases}$$

の下で $w(t_x)+w(t_y)$ を最大にする（x，y）を求め，親ページの境界レコードa，bを t_x，t_y で置き換え，$q_0 t_1 \cdots t_{x-1} q_{x-1}$ をPへ，$q_x t_{x+1} \cdots t_{y-1} q_{y-1}$ をP′へ，$q_y t_{y+1} \cdots t_{4n+k+2}\, q_{4n+k+2}$ をP″へ置く.

なお，Pのすぐ上の兄が満員でその上の兄が満員でない場合も同様である.

4）Pのすぐ下の弟もその次の弟も満員であるとき；



エントリ（S，p_s）とP，P′のスクロール及びQの境界レコードaを綴り合せてできるスクロール

$$q_0 t_1 q_1 \cdots t_{4n+2}\, q_{4n+2}$$

から，条件

$$\begin{cases} n+1 \leq x \leq 2n+1 \\ x+n+1 \leq y \leq x+2n+1 \\ 2n+2 \leq y \leq 3n+2 \end{cases}$$

の下で $w(t_x)+w(t_y)$ を最大にする (x, y) を求め，親ページの境界レコード a を t_y で置き換え，$q_0 t_1 \cdots t_{x-1}\, q_{x-1}$ を P へ，$q_x t_{x+1} \cdots t_{y-1}\, q_{y-1}$ を新しいページ P''' へ，$q_y t_{y+1} \cdots t_{4n+2}\, q_{4n+2}$ を P' へ置き，エントリ (x, p''') を親ページに渡す．（Qに次の段階で (x, p''') が挿入されることになる．即ちこの操作は伝播することになる．）

なお，P のすぐ上の兄もその上の兄も満員である場合も同様である．

(II) レコードの削除

レコードの削除に伴ってページ P のレコードの個数が $2/3\, m$ $(= 4/3\, n)$ を割るときには，高々2枚の隣接した（即ち，すぐ上の2人の兄，又はすぐ下の2人の弟）ページを対象にして，ページ内容の詰め合せを試みる．もし不可能ならば，この3ページと対象に(I)の3)に相当する割り振りを行う．ただし，この場合には $0 \leq k < 2n$ となる．

〈注意〉 McCreight[1]では，可変長レコードを扱うB木について議論している．これは重み付きレコードの特別の場合である．また彼はB*木を，我々の定義1のii)を

ii)′ 根以外の節は $1/2\,m \sim m$ 個のレコードをもち，しかも兄弟が3個以上の節からなる場合には，その兄弟は平均して $2/3\,m$ 個のレコードをもつ．

という条件で置き換えたものに相当する定義をしている．しかも，ページネーションの操作については上記(I)(II)を使っている．しかし，(I)(II)の操作は彼のいうB*木の定義を満足しない．実際に，例はいくらでも作れるが簡単なものとして下図のような例がある．図において，□…□は或る兄弟ページを表わし，□で1/2 fullを■でfullを表わしている．図における最終段階でのページの使用率は，

$$\frac{3+\frac{1}{2}\times 7}{10} = \frac{39}{60} < \frac{40}{60} = \frac{2}{3}$$

→ ■■■

→ ■□□■

→ ■■■■

→ ■■□□■

→ ■■■□■

→ ■■□□□■

⋮

→ ■■□□□□□□□■

となり，彼のB*木の定義における条件ii)′を満足しない．

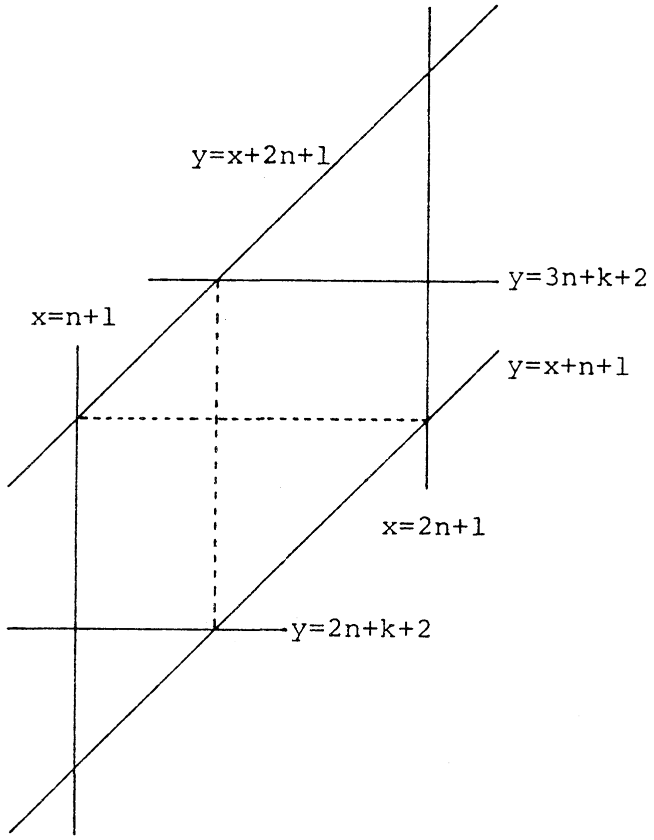
3．ページネーションのアルゴリズム

前節で述べたページネーションは挿入時における3)，4)と削除時における3)に相当する場合を除けば，$O(n)$時間で実行できることは明らかである．3)，4)については素朴な方法では$O(n^2)$時間必要である．McCreight [1]は，ヒープ・ソートを使った$O(n\log n)$時間の方法を示しているが，簡単な工夫によって，$O(n)$時間にちぢめることが示せる．即ち，

<u>定理</u>　重み付きレコードを扱うB*木のページネーションは$O(n)$時間で可能である．

（証明）挿入時における3)，4)及び削除時における3)に相当する場合について考えれば十分である．これら3つの場合は$0\leq k<2n$に対する3)の場合としてまとめられる．そこで，3)における3つの条件式をx，y座標に図示すると次のような六角形の領域が得られる．従って問題はこの領域(境界も含む)内の格子点で$w(x)+w(y)$を最大にするような(x,y)を求める問題になる．この領域における$w(x)+w(y)$の最大値は2つの矩形と2つの三角形における最大値の大きい方の値である．

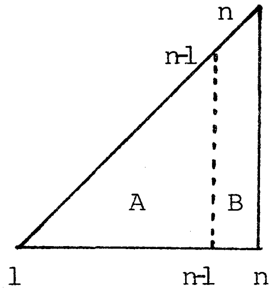
矩形においては



$$\max\{w(x)+w(y);\ a \le x \le b,\ \ c \le y \le d\}$$
$$= \max\{w(x);\ a \le x \le b\} + \max\{w(y);\ c \le y \le d\}$$

であるから，最大値を与える点(x, y)は$O(n)$時間内で求められる.

一方三角形においては，この三角形を下図のように見ると，



$$\max\{w(x)+w(y);\ (x,y)\in A\cup B\}$$
$$=\max(\max\{w(x)+w(y);\ (x,y)\in A\},$$
$$w(n)+\max(\max\{w(y);\ 1\leq y\leq n-1\},\ w(n)))$$

であるから，やはり $O(n)$ 時間で最大値を与える点 (x, y) を求めることができる．□

重み付きレコードを扱う B^* 木を導入し，そのページネーションについて単純で速いアルゴリズムを示した．この B^* 木は McCreight [1] の可変長レコードを扱う B^* 木の拡張になっていることは明らかであろう．ここで考えた重みとしてはレコードの使用頻度などがある．B^* 木を定義するための順序の他に，頻度を考慮して，ページネーションにおける ½ ~ full で平均して ⅔ という自由度を積極的に使って，使用頻度の高いレコードを B^* 木の根になるべく近いところに置こうというわけである．こうすると，外部記憶装置へのアクセス回数を少くすることができる．

参考文献

[1] McCreight, E.: Pagination of B^*-Trees with Variable-Length Records, CACM, 20 (1977), 670-674.

[2] Bayer, R., and McCreight, E.: Organization and

Maintenance of Large Ordered Indexes, Acta Informatica, 1 (1972), 290 - 306.

[3] Comer, D. : The Ubiquitous B-Tree, Computing Surveys, 11 (1979), 121 - 136.