数理解析研究所講究録
第 458 巻 1982 年 11-19

ヒープの解析

山梨大　工学部　計算機科学科
内村桂輔

1. まえがき

データ構造の一つであるヒープは，ヒープソート，([9],[1])として使われるばかりでなくプライオリティキューを作るためにも使われる[5]。

この論文はプライオリティキューのために使われるヒープに新たに一つの要素を挿入する時に必要な交換回数の平均を調べる。

二分木上のヒープの時にはその交換回数の平均は $O(1)$ であることは示されている([2],[3])。ここではその平均値を具体的に求めるとともに t 分木上のヒープについても考察する。

2. 定義

各点が t 個以下の子を持つ木を t 分木という ($t \geq 2$)。

木分木T上の点の数を $|T|$ であらわす。

一般性を失なうことなく我々は次のようなヒープを考えよう。

T上のヒープとは，Tの各点に自然数をつけたもので，次の条件を満たしているもの。

(1)異なる点には異なる数がつけられている。

(2)各点につけられた数はその子達につけられた数より小さい。

我々はヒープに新たに一つの要素を挿入して，再びヒープを作ることを考える。そう入する位置は子の数が木より小の点で深さが最小のもののうちで，一番左側の点の子の位置である。その子のうち空いているものの中で左端の位置。この位置をI.P.ということにする。挿入後，親子の比較，交換をくりかえし(2)の条件が満たされるようにする。この手間の平均的な評価を与えるのが我々の目標である。

Tを木分木として，$|T|=n-1$ とする。このT上に $\{1, 2, \ldots, n\}$ のうちの $n-1$ 個の元でヒープ h を作り，その残りの元を $v(h)$ とする。木分木Tが与えられた時，組 $(h, v(h))$ 全体の集合を H_T であらわし，その要素の数を $|H_T|$ とする。この時 H_T の各組が等確率で生起すると仮定する（[7]参照）。各組 $(h, v(h))$ に対し，$v(h)$ を h に挿入して再びヒー

プにするために必要な交換回数を $b(h)$ とする。この時交換回数の平均 $L(T)$ は次のようになる。

$$L(T) = \Bigl(\sum_{(h, v(h)) \in H_T} b(h) \Bigr) / |H_T| 。$$

3。結果

補題1. T を t 分木として，$|T| = n-1$ とする。T の I.P. から根への距離を d とすると，

$$L(T) = d/n + (n-1) \times R/n,$$

$$R = \sum_{v(h) \neq 1} b(h) / (|H_T| \times (n-1)/n) 。 \qquad \square$$

t 分木 T の部分木で，T の根の子を根とし，T の I.P. を I.P. とするものを $\hat{T}$ とする。すると補題1の R は次のようになる。

補題2.

$$R = L(\hat{T}) \qquad \square$$

補題1と補題2より次の補題が得られる。

補題3. 任意のt分木T（$|T| = n-1$）に対して，TのI.p.の深さをdとする。すると，

$$L(T) = d/n + (n-1) \times L(\hat{T})/n。 \qquad \square$$

t分木T，Sについて，TがSの拡大であるとは，次の三つの条件が成り立つことである。

(1) $|T| = |S| + 1$

(2) TとSの根を一致させた時，SはTの部分木。

(3) TとSのI.p.は一致している。

補題4. t分木T，Sにおいて，TがSの拡大ならば，

$$L(T) < L(S)。 \qquad \square$$

t分木S，Tについて，SがTの短縮形であるとは，次の三つの条件が成り立つことである。

(1) SとTの根を合わせた時，SはTの部分木。

(2) SとTのI.p.は一致する。

(3) SにはI.p.の深さ以上の深さを持つ点はない。

この時補題4より次の補題が得られる。

補題5. t分木S，Tで，SがTの短縮形ならば，

$L(T) < L(S)$。 □

次に完全t分木(complete t-ary tree [6, p. 400])を考える。要素の数がt^n-1の完全t分木を$T(t, n)$であらわす。$t=2$の場合は，補題5より次のことが言える。

<u>補題6.</u> 任意の2分木Tに対して，自然数nが存在して，$L(T) < L(T(2, n))$となる。 □

補題2の前で定義された$\hat{T}$は，Tが$T(t, n)$の場合には，$T(t, n-1)$になることがわかる。故に補題3より次の補題が得られる。

<u>補題7.</u> 任意の$n \geq 2$，$t \geq 2$に対して，

$$L(T(t, n)) = n/t^n + (1 - 1/t^n)\, L(T(t, n-1))。$$

□

次に$1/t$を不定元xでおきかえ，その時の$L(T(t, n))$を$v_n(x)$と書くと，上の式は次のようになる。

$$v_n(x) = nx^n + (1-x^n)\, v_{n-1}(x),\quad (n \geq 2)$$

$$v_1(x) = x。$$

この時，多項式の列 $\{U_n(x)\}$ に関して次の性質が知られている。

定理（[8]）．$U_m(x)$ は $m(m+1)/2$ 次の多項式であり，$U_m(x)=\sum_{n=1}^{m(m+1)/2} a_n^{(m)} x^n$ とおくと次のことが成りたつ。

m 以下の n に対して，

$$a_n^{(m)} = d(n),$$

ここで，$d(n)$ は n の約数の数をあらわす。 □

例えば $U_4(x)$ は次のようになる。

$$U_4(x) = x+2x^2+2x^3+3x^4-3x^5-x^6-2x^7+x^8+2x^9-x^{10}。$$

また，m 以上の n に対しても次の事が成り立つ。

正数 α が与えられた時，自然数 N が存在して，$n \geqq N$ ならば，$|a_n| \leq (1+\alpha)^n$ となる。

$d(n)$ については次の性質が知られている。

定理（[4, p.260]）　任意の正数 δ について，

$$d(n) = O(n^\delta)。$$ □

これらのことより，$K_1(x)=\sum_{n=1}^{\infty} d(n)x^n$とおくと，次のことが成り立つ。

補題8．　$|x|<1$をみたす任意の複素数xについて，

$$\lim_{n\to\infty} |v_n(x)-K_1(x)|=0。$$

□

この事と補題7より$L(T(t,n))$はnに関して単調増加である事がわかる。

定理1．　$L(T(t,n))$は単調増加して，$K_1(1/t)$に収束する。

$K_1(1/t)$を求めるために次のような式の変形を利用する。

$$K_1(x)=\sum_{n=1}^{\infty} d(n)x^n=\sum_{n=1}^{\infty} x^n/(1-x^n)$$
$$=\sum_{n=1}^{\infty} x^{n^2}\left(1+\sum_{k=1}^{\infty} 2x^{kn}\right)$$

この式の変形をもちいて$K_1(1/t)$を小数点以下5ケタ求めたのが次の表である。

t	2	3	4	5
$K_1(1/t)$	1,60669	0,68215	0,42169	0.30175

この事より二分木Tについて，$L(T) < 1.60669$ であることがわかる。

$L(T(t, n))$ の $K_1(1/t)$ への収束のようすは，次の表で与えられる。

t	2	3	4	5
$L(T(t, 10))$	1.59653	0.68206	0.42109	0.30173

文　献

[1] Floyd, R. W., Algorithm 245: Treesort 3, C. ACM, 7, 12, 701 (1964)

[2] Gonnet, G. H. and Rogers, L. D., An algorithmic and complexity analysis of the heap as a data structure, Research Report CS-75-20, Univ. of Waterloo, (1975)

[3] Gonnet, G. H., Heaps applied to event

driven mechanisms, C. ACM, 19, 7, 417-418, (1976)

[4] Hardy, G. H. and Wright, E. M., An Introduction to the Theory of Numbers (Oxford Univ. Press, London, 1960)

[5] Johnson, D. B., Priority queues with update and finding minimum spanning tree, Inf. Proc. Lett., 4, 3, 53 (1975)

[6] Knuth, D. E., Fundamental Algorithm (Addison-Wesley, Reading, Mass., 1968)

[7] Knuth, D. E., Sorting and Searching (Addison-Wesley, Reading, Mass., 1973)

[8] Uchimura, K., An identity for the divisor generating function arising from sorting theory, J. Combinatorial Theory (A) 31, 131-135, (1981)

[9] Williams, J. W. J., Algorithm 232: Heapsort, C. ACM, 7, 6, 347-348 (1964)