論　文

制限つき Deques による順列の生成とソーティング*

今宮淳美** 野崎昭弘**

Abstract

The probrems of generating and sorting permutations using restricted-deques (RDQ) are considered.

Section 2 provides the definitions of operation sequences and the dual correspondence between output RDQ and input RDQ.

Section 3 provides the capabilities of generating and sorting permutations using ORDQ or IRDQ. That is, the necessary and sufficient condition for RDQ to generate or sort the permutation is given.

Then, the relation between the generating and the sorting is given.

Section 4 provides two sorting algorithms using the ORDQ parallel network and the ORDQ cascade network.

1. まえがき

情報処理のための素子または装置のあるクラスが与えられると，それらはどのような能力をもつかと考えるのは当然である．これまでも各種の論理素子と論理関数，オートマタと形式言語の対応など種々の研究がなされてきている．メモリは入出力と記憶する機能をもつ装置（または素子）と考えられる．メモリからの出力は入力の多重集合（重複する記号を含む集合）[2]上の置換である．従ってメモリの与えられたクラスがどのような置換を生成しうるかが興味あるところである．一方，最近計算機アルゴリズムの研究において，メモリのひとつのクラスである線形リストがしばしば用いられている[1),2),5)]．D. E. Knuth は文献 1) で線形リストである Stack の順列生成能力および二進木との対応，文献 2) では Young Tableau との対応等，興味ある結果を示している．Knuth のこれらの指摘により，S. Even と A. Itai による Queues および Stacks の回路による順列の生成とグラフとの関連[3)]や B. Tarjan による Queues および Stacks の回路によるソーティング[4)]が発表されている．これらの場合の入力は多重集合でなく順列（同一の記号を重複して含まない）であり，順列の生成とは，$1,2,\cdots,n$ を順に入力しランダムな順列を出力するものとし，ソーティングとは，ある与えられたランダムな順列を入力し，$1,2,\cdots,n$ の順に出力させる問題である．

ところで線形リストの中で Queue および Stack の一般化となっている Deque および入出力に制限を加えた制限つき Deque に関する研究は，その入出力の一般性が解析を難しくしていて数少ない[1)]．以上の現在までの研究のようすから，本論文では，制限つき Deque の順列生成とソーティング，制限つき Deque からなる回路によるソーティング・アルゴリズムが示される．

まず **2.** では，本論文で必要となる諸定義および，制限つき Deque の入出力操作に関する性質が示される．**3.** では，各制限つき Deque による順列生成とソートとの関連および制限つき Deque 間の対応が示される．**4.** では，出力制限つき Deque からなる並列回路，直列回路，各々によるソーティング・アルゴリズムが示される．

2. 準　備

この章では，以下の議論に必要となる諸定義，および二種類の制限つき Deque の入出力操作間の関係を

* Generating and Sorting Permutations Using Restricted-Deques by Atsumi IMAMIYA and Akihiro NOZAKI (Department of Computer Science, Yamanashi University)

** 山梨大学工学部計算機械学科

示す.

〔**定義 1**〕[1] 線形リストは点の一次元相対位置だけを含む点 $L[1], L[2], \cdots$, の集合である. ただし $L[1]$ が一番目の点で $1<k$ のとき k 番目の点 $L[k]$ は $L[k-1]$ の後, $L[k+1]$ の前に位置する. ☒

〔**定義 2**〕[1] Deque (Double-ended queue) とは, すべてのデータの積みこみ (Loading), 積みおろし (Unloading) がリストの両端でなされる線形リストである. ☒

〔**定義 3**〕[1] 出力制限 (入力制限) Deque とは, 積みおろし (積みこみ) が, リストの定められたひとつの端だけでなされる制限つき Deque である. ☒

出力制限 (入力制限) Deque では入力(出力)は両方向からなされる. 以下では, 制限つき Deque を RDQ (**R**estricted **Deque**), 出力制限 (入力制限) Deque を O(I)RDQ (**O**utput (**I**nput) **R**estricted **Deque**) と略記する.

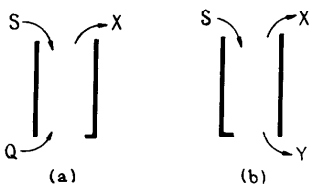
ORDQ において, リスト各端からの積みこみを S と Q, 積みおろしを積みこみの S と同じ端から行いそれを X で各々表わす. IRDQ の積みこみを S, 各端からの積みおろしを X と Y で表わす. ただし, X は積みこみの S と同じ端からであるとする (**Fig. 1** 参照).

便宜上, 各 RDQ 中にある二記号の間で, 積みおろし X によって, より以前に積みおろし可能な記号は上部, 後に積みおろし可能な記号は下部にあるということにする.

ORDQ の長さ n の順列に対する操作列 Θ とは, いくつかの S, Q および X からなる長さ $2n$ の列であり, その列全体では S と Q の個数の和, X の個数も n である. $\Theta=\theta_1, \theta_2, \cdots, \theta_{2n}$ において $SQ(\theta_j), X(\theta_j)$ を次の様に定義する.

$$SQ(\theta_j)=\begin{cases}1 \Leftarrow \theta_j=S, Q,\\ 0 \Leftarrow \theta_j=X,\end{cases}$$

$$X(\theta_j)=\begin{cases}1 \Leftarrow \theta_j=X,\\ 0 \Leftarrow \theta_j=S, Q.\end{cases}$$



(a) Output-Restricted Deque (ORDQ),
(b) Input-Restricted Deque (IRDQ).

Fig. 1 Restricted Deques

任意の i $(1\leq i\leq 2n)$ に対して,

$$\sum_{1\leq j\leq i} SQ(\theta_j)\geq \sum_{1\leq j\leq i} X(\theta_j) \qquad (1)$$

である場合 (そのときに限る), この操作列 Θ は**許容列**であるという.

同様に IRDQ に関する操作列 Γ とは, いくつかの S, X および Y からなる長さ $2n$ の列であり, その列全体では S の個数は n, X と Y の個数の和も n である.

$\Gamma=\gamma_1\gamma_2\cdots\gamma_{2n}$ において $S(\gamma_j), XY(\gamma_j)$ を次のように定義する.

$$S(\gamma_j)=\begin{cases}1 \Leftarrow \gamma_j=S,\\ 0 \Leftarrow \gamma_j=X, Y,\end{cases}$$

$$XY(\gamma_j)=\begin{cases}1 \Leftarrow \gamma_j=X, Y,\\ 0 \Leftarrow \gamma_j=S.\end{cases}$$

任意の i $(1\leq i\leq 2n)$ に対して,

$$\sum_{1\leq j\leq i} S(\gamma_j)\geq \sum_{1\leq j\leq i} XY(\gamma_j) \qquad (2)$$

である場合 (そのときに限る), この操作列 Γ は許容列であるという.

一般に許容列 Δ に従って順列 P を ORDQ または IRDQ に対し, 積みこみ, 積みおろしを行って出力として順列 R が得られた場合,

$$\Delta(P)=R \qquad \text{と表わす.}$$

ここで, $P=1, 2, \cdots, n$ の場合に**順列の生成**, $R=1, 2, \cdots, n$ の場合に**順列のソーティング**という.

一般に, 操作列と生成 (ソート) される順列との間には一対一対応はない. 例えば ORDQ について, $P=12$ のとき $SXSX$ と $QXQX$ は両方とも順列 $R=12$ を生成する. しかし Knuth は ORDQ について, 次の制限を操作列に加えることにより生成される順列との間に一対一対応があることを示した[1].

(i) ORDQ が空の場合, 積みこみに Q を用いる.

(ii) XQ は許さず, QX を用いる.

ここで ORDQ と IRDQ に関する操作列間の対応を示す. まず上記の(i), (ii)と同様に, IRDQ についても次の命題に示す制限を加えることによって, 操作列と生成される順列間に一対一対応が示される.

〔**命題 1**〕 IRDQ に関する操作列に次の制限を加えると生成される順列との間に一対一対応がある.

(i)′ IRDQ が一個の記号だけを含む場合, その記号の積みおろしは Y によって行う.

(ii)′ YS は許さず, SY を用いる.

証明は, 制限(i)′, (ii)′ のもとで, 二つの異なる操作列が同じ順列を生成すると仮定し, 矛盾することを

示す．証明略．　⊠

ORDQ に関する操作列 Θ に対して IRDQ に関する操作列 Θ^* を次のように対応させる：$1\leq i\leq 2n$ なるすべての i に対して，

$$\theta_i{}^*=\bar{\theta}_{2n+1-i},$$

ただし，$1\leq j\leq 2n$，

$$\bar{\theta}_j=\begin{cases}S\Leftrightarrow\theta_j=X,\\ X\Leftrightarrow\theta_j=S,\\ Y\Leftrightarrow\theta_j=Q.\end{cases}$$

すなわち，Θ を右から読み X を S，S を X，Q を Y に代えて左から順に書き Θ^* を作る．たとえば，

$\Theta=QSQSXXXX$　　ならば，

$\Theta^*=SSSSXYXY$　　となる．

この場合，上記の対応関係にある Θ と Θ^* は互いに**双対**であるという．$(\Theta^*)^*=\Theta$ が成り立つ．

〔**命題2**〕 ORDQ に関する Θ が許容列であるための必要十分条件は，IRDQ に関する双対対応 Θ^* が許容列であること．

証明は（1），（2）式と双対対応からなされる．証明略．　⊠

〔**命題3**〕 $\Theta_i(I)=\Theta_k(I)$ であるための必要十分条件は，$\Theta_i{}^*(I)=\Theta_k{}^*(I)$，$I=1,2,\cdots,n$．

証明は命題1と双対対応からなされる．証明略．　⊠

3. RDQ による順列の生成とソーティング

この章では，まず各 RDQ による順列生成能力を示す．次に ORDQ と IRDQ について，順列の生成とソーティングの関係，双対対応により生成，ソートされる順列の関係を示す．

3.1 RDQ の順列生成能力

RDQ による順列の生成において，ある時点で記号 P を RDQ から積みおろしたいのに記号の積まれた状態によってそれができない場合，この RDQ は**閉鎖**するということにする．いちど積みおろされた記号は再び積みこまれることはないものとする．

〔**定義4**〕 $1\leq i_1<\cdots<i_k\leq n$ のとき，$p_{i1},p_{i2},\cdots,p_{ik}$ は $p_1,p_2,\cdots,p_n$ の部分列とよばれる．　⊠

各 RDQ の順列生成能力は次の二つの定理で示される．

〔**定理1**〕 ORDQ が入力列 $1,2,\cdots,n$ を出力列 $p_1,p_2,\cdots,p_n$ に置換できるための必要十分条件は，出力列 $p_1,p_2,\cdots,p_n$ が，次の（1型）または（2型）の条件をみたす部分列 $p_{i1},p_{i2},p_{i3},p_{i4}$ を含まないことである：$1\leq i_1<i_2<i_3<i_4\leq n$，

（1型）　$p_{i1}>p_{i3}>p_{i4}>p_{i2}$，

（2型）　$p_{i1}>p_{i3}>p_{i2}>p_{i4}$．

（証明）　必要性：　順列（出力列）$p=p_1,p_2,\cdots,p_n$ が（1型）をみたす部分列 $p_{i1},p_{i2},p_{i3},p_{i4}$ を含むと仮定する．（1型）の条件から p_{i1} が ORDQ に積みこまれて，おろされるまで p_{i2},p_{i3},p_{i4} は，共に積みおろされることはない．順列 p が生成されるためには，p_{i1} が積みこまれた時点で **Fig. 2**（a）のように積まれていなければならない．（1型）の条件から $p_{i1},p_{i2},p_{i3},p_{i4}$ の四記号間では，p_{i2} がはじめに S または Q の方向から積みこまれる．出力は $p_{i1},p_{i2},p_{i3},p_{i4}$ の順であるから，次に p_{i4} を Q の方向から ORDQ に積みこまなければならない（Fig. 2（b））．その後に，p_{i2} と p_{i4} の間に p_{i3} を入れなければならないが，これは Fig. 2（b）の状態から不可能である．従って順列 p は ORDQ で生成されない．（2型）についても同様．

十分性：　いまある記号を積みおろさなければならないのに，その上部にいくつかの記号があって，ORDQ が閉鎖すると仮定する．閉鎖すれば順列は生成されないし，その逆も成り立つから閉鎖する場合の条件を求めてやれば，その条件をみたさない場合が，その順列を生成できるための十分条件となる．

次のアルゴリズムに従って記号の積みこみ，積みおろしを行う．“いま $p_1,\cdots,p_k$ まで出力されているとする．p_{k+1} が ORDQ の最上部の記号となまるで，入力記号を順次，次のように積みこむ．記号 i を積みこむ時点での ORDQ の最上部の記号を u とする．出力順列 P において，記号 i が u より左側にある（すなわち以前に積みおろされる）ならば S の方向，右側にあるならば Q の方向より積みこむ．p_{k+1} が，ORDQ の最上部の記号となった時点で積みおろす．”

では，上記のアルゴリズムで生成されない順列は，どのような条件をもつであろうか．いま記号 a を積み

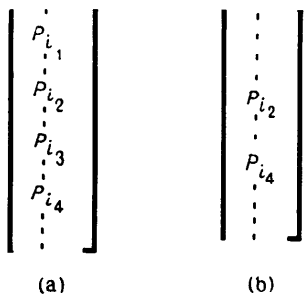


Fig. 2　Aspect of Symbols Loaded in ORDQ

おろそうとする時点で ORDQ が閉鎖すると仮定する．すなわち，a の上部には，少なくとも一記号 b が存在して出力順列において a は b より左側にある．このような b が存在するための条件を以下で求める．

(1) $a>b$: a が積みこまれる時点での ORDQ の最上部の記号 $u(a>u, u\neq b^*)$ は出力順列では a より左側にある*.

従って，a を Q の方向から ORDQ に積みこむ．a, b, u はある記号 c が (S の方向から) 積みこまれ，おろされるまで ORDQ 中に共になければ a の積みおろしの時点で ORDQ が閉鎖することにはならない．従って $c>a, b, u$.

いま　$c=p_{i1}, u=p_{i2}, a=p_{i3}, b=p_{i4}$

とおけば，$1\leq i_1<i_2<i_3<i_4\leq n$,

(1型)　$p_{i1}>p_{i3}>p_{i4}>p_{i2}$,

(2型)　$p_{i1}>p_{i3}>p_{i2}>p_{i4}$,

のどちらかをみたす．

(2) $a<b$: b が積みこまれる時点で ORDQ の最上部に記号 v があるとする $(b>v, v\neq a^{**})$.

仮定により，出力順列において b は v より左側になければ，アルゴリズムにより a の上部に積みこまれない．このような a, b, v が ORDQ 中に同時に存在するためには(1)の場合と同様に，これらの記号の上部に記号 c が存在し，$c>a, b, v$ をみたす．$c=p_{i1}, a=p_{i2}, b=p_{i3}, v=p_{i4}$ とおけば，$1\leq i_1<i_2<i_3<i_4\leq n$,

(1型)　$p_{i1}>p_{i3}>p_{i4}>p_{i2}$,

(2型)　$p_{i1}>p_{i3}>p_{i2}>p_{i4}$.

のどちらかをみたす．

以上(1), (2)より順列が(1型), (2型)をみたす部分列を含まなければ，その順列は ORDQ によって生成される．　⊠

定理1と同様の考察により，IRDQ について次の定理が成り立つ (証明略)．

〔**定理2**〕 IRDQ が入力列 $1, 2, \cdots, n$ を出力列 $p_1, p_2, \cdots, p_n$ に置換できるための必要十分条件は，出力列 $p_1, p_2, \cdots, p_n$ が，次の(2型)または(3型)の条件をみたす部分列 $p_{i1}, p_{i2}, p_{i3}, p_{i4}$ を含まないことである．$1\leq i_1<i_2<i_3<i_4\leq n$,

(2型)　$p_{i1}>p_{i3}>p_{i2}>p_{i4}$,

(3型)　$p_{i1}>p_{i4}>p_{i2}>p_{i3}$.　⊠

* そうでなければ，上記のアルゴリズムに従って a を S の方向から積みこめて，b が a の上部にあるとする仮定に反する．

** そうでなければ，上記のアルゴリズムによりの積みおろしの時点で閉鎖するという仮定に反する．

** a は恒等置換をあらわす．

3.2 順列生成とソーティング

この小節では，命題の証明の便宜上，順列の生成とソーティングを次のように定義する．

$\{1, 2, \cdots, n\}$ 上の順列 $P=p_1, p_2, \cdots, p_n$ に対して，**逆順列** $P^{-1}=q_1, q_2, \cdots, q_n$ を定義する．

$$P^{-1}=\begin{pmatrix}1 & 2\cdots n\\ q_1 & q_2\cdots q_n\end{pmatrix}=\begin{pmatrix}1 & 2\cdots n\\ p_1 & p_2\cdots p_n\end{pmatrix}^{-1}.$$

〔**定義4**〕 P が I(O)RDQ で**生成可能**⇔I(O)RDQ に対する許容列 θ に関して，$a_1, a_2, \cdots, a_n$ を入力すると $a_{P(1)}, a_{P(2)}, \cdots, a_{P(n)}$ が得られる．ここに，$P(i)$ は $\{1, 2, \cdots, n\}$ 上の順列 P の第 i 成分である．

〔**定義5**〕 P が I(O)RDQ で**ソート可能** ⇔ I(O)RDQ に対する許容列 θ に関して，$a_{P(1)}, a_{P(2)}, \cdots, a_{P(n)}$ を入力すると $a_1, a_2, \cdots, a_n$ が得られる．

ここでは **2.** と同様に許容列 θ の双対を θ^* であらわす．

$$\tau=\begin{pmatrix}1 & 2 & \cdots n-1 & n\\ n & n-1\cdots & 2 & 1\end{pmatrix}\text{ とおく．}$$

$\tau^{-1}=\tau$, $\tau^2=e^{***}$

〔**補題1**〕 順列 P が I(O)RDQ で生成可能であるための必要十分条件は，P^{-1} が I(O)RDQ で (同じ許容列により) ソート可能であること．

(証明) 順列の生成とソートの対応から一方向のみを証明すれば，逆方向の証明は同様である．

ある許容列 θ に関して，$a_1, a_2, \cdots, a_n$ を入力して，$a_{P(1)}, a_{P(2)}, \cdots, a_{P(n)}$ が得られたとする．当然，$b_1, b_2, \cdots, b_n$ を入力すれば，$b_{P(1)}, b_{(2)}, \cdots, b_{P(n)}$ が得られる．$1\leq i\leq n$ なるすべての i について，$b_i=a_{P^{-1}(i)}$ とおくと，

$a_{P^{-1}(1)}, a_{P^{-1}(2)}, \cdots, a_{P^{-1}(n)}$ を入力して，

$a_1, a_2, \cdots, a_n$ が得られる．ただし，$P^{-1}(i)$ は P の逆順列の第 i 成分．　⊠

定理1, 2と補題1より，次の定理が成り立つ．

〔**定理3**〕 O(I) RDQ で順列 $P=q_1, q_2, \cdots, q_n$ が，ソートされるための必要十分条件は，P が条件(1型)′ または (2型)′ ((2型)′ または (3型)′) をみたす部分列 $q_{i1}, q_{i2}, q_{i3}, q_{i4}$ を含まないこと．

(1型)′　$q_{i2}>q_{i3}>q_{i1}>q_{i4}$,

(2型)′　$q_{i1}>q_{i3}>q_{i2}>q_{i4}$,

(3型)′　$q_{i3}>q_{i1}>q_{i2}>q_{i4}$.

(略証) ORDQ の(1型)′について示す．他も同様に示される．定理1の条件が補題1に示した P^{-1} でどのように変換されるかを示せばよい．P が ORDQ でソートされないと仮定する．補題1より，P^{-1} は OR-

DQ で生成されない.

$$P^{-1}=\begin{pmatrix}\cdots j_1\cdots j_2\cdots j_3\cdots j_4\cdots\\ \cdots p_{j1}\cdots p_{j2}\cdots p_{j3}\cdots p_{j4}\cdots\end{pmatrix},$$

ただし, $j_1<j_2<j_3<j_4$, と表わされる.

P^{-1} が定理1に示す(1型)をみたす部分列をもつとする. この場合, P は次のように表わされる.

$$P=(P^{-1})^{-1}\begin{pmatrix}\cdots p_{j2}\cdots p_{j4}\cdots p_{j3}\cdots p_{j1}\cdots\\ \cdots j_2\cdots j_4\cdots j_3\cdots j_1\cdots\end{pmatrix}$$
$$=\begin{pmatrix}\cdots i_1\cdots i_2\cdots i_3\cdots i_4\cdots\\ \cdots q_{i1}\cdots q_{i2}\cdots q_{i3}\cdots q_{i4}\cdots\end{pmatrix}.$$

$j_1<j_2<j_3<j_4$ より

(1型)′　$q_{i2}>q_{i3}>q_{i1}>q_{i4}$.　⊠

〔**補題2**〕 順列 P が I(O)RDQ で生成可能であるための必要十分条件は, $\tau P\tau^{-1}$ が O(I)RDQ でソート可能であること. ただし, $P=p_1, p_2\cdots, p_n$ のとき, $\begin{pmatrix}1 & 2 & \cdots n-1 & n\\ n & n-1\cdots & 2 & 1\end{pmatrix}\cdot\begin{pmatrix}1 & 2\cdots n\\ p_1 & p_2\cdots p_n\end{pmatrix}\cdot\begin{pmatrix}1 & 2 & \cdots n-1 & n\\ n & n-1\cdots & 2 & 1\end{pmatrix}=\begin{pmatrix}1 & 2\cdots n\\ b_1 & b_2\cdots b_n\end{pmatrix}$ であるならば, $\tau P\tau^{-1}$ は順列 $b_1, b_2, \cdots, b_n$ をあらわす.

(証明) **Fig. 3** と双対対応から次の命題が成り立つ. IRDQ に $a_1, a_2, \cdots, a_n$ を入力して, 許容列 Θ によって, $a_{P(1)}, a_{P(2)}, \cdots, a_{P(n)}$ が生成される. ⇔ORDQ に $b_{P(n)}, \cdots, b_{P(1)}$ を入力して許容列 Θ^* によって, $b_n, \cdots, b_1$ が得られる. そこで, $b_j=a_{\tau(j)}$, $1\leq j\leq n$, とおく.

$$b_{P(n)}=a_{\tau(P(n))}=a_{\tau P\tau^{-1}(1)}, \cdots, b_{P(1)}=a_{\tau P\tau^{-1}(n)}.$$

ただし, $\tau P\tau(i)$ は, $\tau P\tau^{-1}$ が表わす順列の第 i 成分である.

すなわち, $a_{\tau P\tau^{-1}(1)}, \cdots, a_{\tau P\tau^{-1}(n)}$ を入力して, $a_1, a_2, \cdots, a_n$ が得られる. ゆえに $\tau P\tau^{-1}$ が ORDQ でソート可能. 同様に逆も成り立つ. ⊠

補題1,2より, 双対な許容列で生成される順列について, 次の定理が成り立つ.

〔**定理4**〕 IRDQ に $1, 2, \cdots, n$ を入力して, 許容列 Θ によって順列 P が得られ, ORDQ に $1, 2, \cdots, n$ を入力して許容列 Θ^* によって順列 P^* が得られたとする. $P=P^*$ であるための必要十分条件は,

$$P\tau=\tau P^{-1}.$$

(証明) ORDQ で P^* が生成可能 (Θ^* による). $\overset{①}{\Longleftrightarrow}$ORDQ で $(P^*)^{-1}$がソート可能 (Θ^* による). $\overset{②}{\Longleftrightarrow}$ IRDQ で $\tau(P^*)^{-1}\tau^{-1}$ が生成可能 (Θ による). ①は補題1, ②は補題2による.

IRDQ で許容列 Θ によって, P が生成されると仮定しているから, ①, ② より $P=\tau(P^*)^{-1}\tau^{-1}$. 定理の条件より $P=P^*$ であるから

$$P\tau=\tau P^{-1}\quad \text{が成り立つ.}$$

同様に逆も成り立つ. ⊠

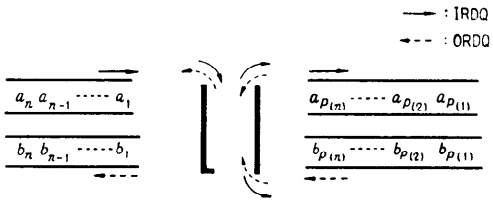


Fig. 3 The Relation between Generating by ORDQ and Sorting by IRDQ.

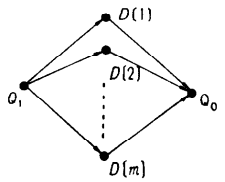
4. ORDQ 回路によるソーティング・アルゴリズム

線形リストからなる回路によるソーティングについては, Queue と Stack に関して R. Tarjan, D. Knuth らの研究がある[2),4)]. しかし, それらの研究では, Deque および RDQ の回路によるソーティングのアルゴリズムなど明らかにされていない点が多い. この章では, ORDQ からなる並列回路, 直列回路によるソーティング・アルゴリズムを示す.

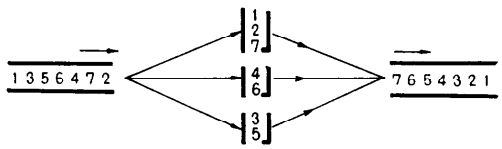
4.1 ORDQ 並列回路によるソーティング

Fig. 4 (a) に示す ORDQ 並列回路を用いて, Q_I (入力用 Queue) にある順列 $P=p_1, p_2, \cdots, p_n$ を各 ORDQ, $D[k]$, を介して Q_0 (出力用 Queue) に 1, 2, $\cdots$, n の順に積みおろすソーティング・アルゴリズムを示す. 各 ORDQ 間では記号のやりとりはない.

ソートすべき順列の部分列に関して, 次の集合を定



(a) ORDQ Parallel Net



(b) Sorting 2746531 Using ORDQ Parallel Net

Fig. 4 Sorting by ORDQ Parallel Net.

義する.

$c_{[j]}{}^{q_{i4}}=\{(q_{i1},q_{i2},q_{i3})\mid$部分列 $q_{i1}q_{i2}q_{i3}q_{i4}$ が定理3に示す$(j)'$型をみたす.$\}$, $j=1,2$.

$$c_{[j]}=\bigcup_{p_{i4}} c_{[j]}{}^{q_{i4}},\ c[0]=c[1]\cup c[2].$$

〔**性質1**〕 順列が並列回路でソートされるためには, $c_{[j]}\ni(q_{i1},q_{i2},q_{i3})$ なる記号 q_{i1},q_{i2},q_{i3} が三記号とも同一の ORDQ 中に積みこまれることはない.

(証明) $c_{[j]}\ni(q_{i1},q_{i2},q_{i3})$であれば, $c_{[j]}{}^{q_{i4}}\ni(q_{i1},q_{i2},q_{i3})$ なる q_{i4} が存在する. $j=1$ の場合を考える. $j=2$ の場合も同様である.

q_{i1},q_{i2},q_{i3} ともに同一の ORDQ $D[i]$ に積みこまれると仮定する. $q_{i2}>q_{i3}>q_{i1}>q_{i4}$ であるから, q_{i4} が Q_1 からある $D[k]$ に積みこまれる以前に q_{i1}, q_{i2},q_{i3} はいずれも $D[i]$ 中にある. さらに q_{i1},q_{i2}, q_{i3} が, Q_0 に積みおろされる以前に q_{i4} は積みおろされねばならない. 従って **Fig. 5** の $D[i]$ において, $(a_1,a_2,a_3)=(q_{i2},q_{i3},q_{i1})$ でなければならないが, 実際には $(a_1,a_2,a_3)=(q_{i1},q_{i2},q_{i3})$, (q_{i3},q_{i1},q_{i2}), (q_{i2},q_{i1},q_{i3}), (q_{i3},q_{i2},q_{i1}). 従って, この場合ソートできない. ☒

上記の〔性質1〕を用いて, 並列 ORDQ 回路によるソーティング・アルゴリズム PS を示す.

アルゴリズム PS: 順列 $P=p_1,p_2,\cdots,p_n$ について $c[0]=\{\bar{c}_1,\bar{c}_2,\cdots,\bar{c}_t\}$ が与えられているとする. ただし, $\bar{c}_k=(q_{i1},q_{i2},q_{i3})$. 記号 a が $D[i]$ 中にある場合, $a\in D[i]$ と表わす.

PS 1: $j\leftarrow 1$

PS 2: $i\leftarrow 1$

PS 3: $\forall a,b\in D[i]$ に対して, $c[0]$ 中に $(a,b,p_j)\in\bar{c}_k$ なる $\bar{c}_k$ が存在すれば, $i\leftarrow i+1$ として PS 3 を行う. $\bar{c}_k$ が存在しなければ PS 4 へ.

PS 4: $D[i]\Leftarrow D[i]\oplus\{p_j\}$*

PS 5: $j\leftarrow j+1$,

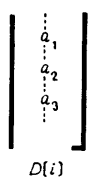


Fig. 5 Aspect of Symbols Loaded in of ORDQ Parallel Net.

* ⊕の意味: P_j を積みこむ直前における $D[i]$ の最上部の記号 a に対して, $a<P_j$ ならば, Q の方向, $a>P_j$ ならば S の方向から P_j を Di に積みこむ.

** $\lceil X\rceil$ は, X かまたはそれを越える最小整数を表わす.

$j\leq n$ であれば PS 2 へ, $j>n+1$ であれば, 積みこみ終了し, 各 ORDQ の最上部の記号を比較して, 最も小さい記号の順に積みおろす. ☒

〔**例題1**〕 順列 2746521 をアルゴリズム PS を用いてソートする: $c[0]=\{(2,7,1),(2,7,6),(2,7,5),(2,7,3),(2,4,3),(2,6,5),(2,6,3),(2,5,3),(4,6,5),(4,6,3),(4,5.3),(7,4,6)\}$. 積みこみの結果を Fig. 4(b) に示す. ☒

4.2 ORDQ 直列回路によるソーティング

この小節では, **Fig. 6**(a)に示す ORDQ 直列回路による順列のソーティング・アルゴリズムと必要とする ORDQ の個数を示す. 直列回路の各段の ORDQ に n 個の記号がすべて前段から積みこまれ, そのあとで次の段の ORDQ に積みおろされる.

アルゴリズム CS: ORDQ $D[k]$ に積まれている記号を $[p(1),p(2),\cdots,p(n)]_k$ で表わす. ただし, $i_1<i_2$ ならば $p(i_1)$ は $p(i_2)$ の上部にあるものとする. $[p(1),p(2),\cdots,p(n)]_0=[p_1,p_2,\cdots,p_n]_0$ とする. $k=0$, $2\lceil\log_2 n\rceil+1$ なる $D[k]$ は Queue と考える. 各 $p(i)$ は二進数で表示される.

$$p(i)=b_p\cdot 2^{p-1}+b_{p-1}\cdot 2^{p-2}+\cdots+b_2\cdot 2+b_1.$$

ただし $p=\lceil\log_2 n\rceil$**, $b_j=0,1$.

CS 1: $k\leftarrow 0$, $j\leftarrow 0$.

CS 2: $k\leftarrow k+1$, $j\leftarrow j+1$, $i=1,2,\cdots,n$ の順に次の操作を行う; $b_j=0$ ならば S の方向, $b_j=1$ ならば Q の方向から各々 $D[k]$ に $p(i)$ を積みこむ.

CS 3: $k\leftarrow k+1$, $j\leq p$ ならば, $i=1,2,\cdots,n$ の順に次の操作を行う; $D[k-1]$ より各 $p(i)$ を積みおろし, $b_j=0$ ならば S の方向, $b_j=1$ ならば Q の方向から各々 $D[k]$ に $p(i)$ を積みこむ. そのあと **CS 2** へ. $j>p$ ならば **CS 4** へ.

CS 4: アルゴリズム終了. $k=2\lceil\log_2 n\rceil$. $D[2p]$ から $D[2p+1]$ へ Q の方向から積みこむ. ☒

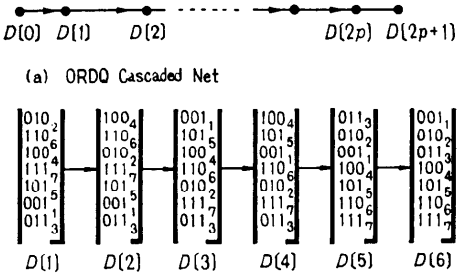


Fig. 6 Sorting by ORDQ Cascaded Net.

上記のアルゴリズム CS が任意の順列をソートすることの略証は次のとおりである（詳しくは文献 6) の $p.63$ を参照されたい）．アルゴリズム CS でソートされない二つの数が存在すると仮定する．すなわち $D[2\lceil\log_2 n\rceil]$ 中で，$i_1<i_2$ かつ $P(i_1)>P(i_2)$ なる二つの数 $P(i_1)$ と $P(i_2)$ が存在するとする．$p(i_1)$ と $p(i_2)$ を二進数表示した場合，各ビットを $(b_q)i_1, (b_q)i_2$，$1\leq q\leq\lceil\log_2 n\rceil$，と略記する．この場合，上記の二つの数で異なるビットの最大の桁を j とする．すなわち $(b_j)i_1=1, (b_j)i_2=0$．仮定とアルゴリズム CS により $D[2j-1]$ では，$p(i_2)$ は $p(i_1)$ より上部に積まれている．次にアルゴリズム CS の CS 3 によって，$(b_j)i_2=0$ より $p(i_2)$ が S の方向より，$(b_j)i_1=1$ より $p(i_1)$ が Q の方向より各々 $D[2j]$ に積みこまれる．$j+1$ 桁以上のビットでは，$p(i_1)$ と $p(i_2)$ の各 b_q は一致するとしているから，$D[2j+1]$ 以上の段では，$p(i_1)$ と $p(i_2)$ は同一の方向より積みこまれる．$j+1\leq l\leq\lceil\log_2 n\rceil$ なる $D[2l]$ においては，アルゴリズム CS により $p(i_2)$ が $p(i_1)$ の上部にあることになる．従って，$D[2\lceil\log_2 n\rceil]$ において $p(i_1)$ が $p(i_2)$ より前に出力されるとした仮定に矛盾する．ゆえに，アルゴリズム CS は任意の順列をソートすることができる． ☒

〔**例題 2**〕 順列 7465132 をアルゴリズム CS を用いてソートする．結果は Fig. 6(b) に示される． ☒

〔**定理 5**〕 アルゴリズム CS は，$2\lceil\log_2 n\rceil$ 個の ORDQ からなる直列回路で長さ n の任意の順列をソートする． ☒

4.3 アルゴリズムの検討

アルゴリズム PS は，ORDQ の間の記号の受け渡しがなく，各 ORDQ がある個数以下の記号しか積みこめない場合に対応している．この場合，三項列の集合 $\bar{c}_k$ を見つけるアルゴリズムが必要となる（文献 7) 参照）．

アルゴリズム CS は，ラディクス・ソートの一種である．読者は，ORDQ 間の記号の受け渡しに注意されたい．ソートすべき順列中に，大きさが非常にはなれている数字が存在する場合には，アルゴリズム CS は冗長な点がみられる．

5. すむび

本論文では線形リストである制限つき Deque について，順列の生成能力とソーティングに関していくつかの結果を得た（定理 1, 2, 3）．また順列の生成とソーティングの関係，ORDQ 回路による二種類のソーティング・アルゴリズムを示した（これらは，いずれも同様なアルゴリズムで IRDQ 回路でも成り立つ）．

未解決の問題：

（1） 与えられた順列が，ある部分列（定理 1, 2, 3）を含むかどうか，含むならばすべて列挙する有効なアルゴリズムを示すこと．

（2） **4.1** の並列回路によるソーティングで必要とする ORDQ の個数（順列の長さの関数）．

（3） Deque の順列生成能力およびソーティング．

謝辞 日頃，御指導下さる東北大学，大泉，野口，斉藤教授ならびに各研究室，山梨大学有沢助教授に感謝します．また種々の有益な指摘をされた査読者に感謝します．

参考文献

1) D. E. Knuth: Fundamental Algorithms, Addison Weseley, Ch. 2. (1968).
2) D. E. Knuth: Sorting and Searching, Addison Weseley, Ch. 5. (1973).
3) S. Even and A Itai: Queue, Stacks, and Graphs, Theory of Machines and Computations, Academic Press, pp. 71～86.
4) R. Tarjan: Sorting Using Networks of Queues and Stacks, J. ACM, Vol. 19, No. 2, pp. 341～346. (1972).
5) A. V. Aho, J. E. Hopcroft, J. D. Ullman: The Design and Analysis of Computer Algorithms, Addison Weseley, (1974).
6) 今宮：制限つき Deque システムによる順列生成とソーティング，電子通信学会技報，AL 75-61, pp. 57～64. (1975-12).
7) 今宮：制限つき Deque で生成されない順列の同定アルゴリズム，電子通信学会技報，AL 75-62, pp. 65～72. (1975-12).

（昭和 51 年 1 月 19 日受付）
（昭和 51 年 6 月 14 日再受付）

訂　　正

昭和51年12月号掲載の論文「制限つき Deques による順列の生成とソーティング」今宮淳美，野崎昭弘に次のような訂正個所があります．

- p. 1128　左脚注

  計算機械学科 ⇒ 計算機科学科

- p. 1131　左脚注

  ** a は ⇒ e は

- p. 1132　左↓6

  $P=(P^{-1})^{-1}\begin{pmatrix}\cdots\\ \cdots\end{pmatrix} \Rightarrow P=(P^{-1})P^{-1}=\begin{pmatrix}\cdots\\ \cdots\end{pmatrix}$

- p. 1133　右↓4

  順列 2746521 ⇒ 順列 2746531

- p. 1133　右↓5

  (2, 7, 1) ⇒ (2, 7, 4)

- p. 1133　右↓6～7

  (4, 6, 3), (4, 5, 3) ⇒ 消去

- p. 1134　右↓15

  識辞 ⇒ 謝辞

p. 1134　右↑15

pp. 71～86 ⇒ pp. 71～86 (1971)