Journal of Mathematical Research and Exposition
Vol.11 No.1 Feb. 1991

Computational Equivalence Between the Problems of Sorting and Monotone Reconstruction for Random Permutations*

Kequan Ding

(Department of Mathematics, University of Wisconsin-Madison Madison, WI 53706, USA)

Abstract

In this note, it is shown that the monotone reconstruction problem is equivalent to that of sorting, in the sense of computational complexity. In particular from any given sorting algorithm A, an algorithm B for the monotone reconstruction problem can be developed with at most $O(m)$ time and $O(m)$ space cost more than that used in A, and vice versa. As a consequence of this result, it is obtained that the time complexity of the monotone reconstruction problem of n-element random permutations is $O(n\log n)$.

I. Introduction

Let $[n]=\{1,2,\cdots,n\}$. An repeatable sequence $\pi: a_1 a_2 \cdots a_m$ with $a_i \in [n]$ is calledan (m,n)-random permutation if each element of $[n]$ appears in π at least once. Let $S(m,n)$ be the set of all (m,n)-random permutations. Clearly, $S(n,n)$ is the set of all n-element permutations without repeatitions. Let P be any given set of properties satisfied by a subset of $S(m,n)$, denote the subset by $S(m,n;P)$, i.e.,

$$S(m,n;P)=\{\pi \in S(m,n) \mid \pi \text{ satisfies } P\}.$$

The minimal partition problem of random permutations (which is also called reconstruction problem, for the sake of simplicity) can be formulated in the following way: for a given set P of properties on $S(m,n)$, $\forall \pi \in S(m,n)$, find an element $\pi' \in S(m,n;P)$ such that

1) $\pi'=\pi_1 \circ \pi_2 \circ \pi_3 \circ \cdots \circ \pi_k$, where "$\circ$" denotes the juxtaposition operation,

2) each π_i is a subsequence of π,

3) $\{\pi_i \mid 1 \le i \le k\}$ is a partition of the set of terms of π,

4) the number k is minimal.

Usually $\cdots \pi(i)$ denotes the i-th term a_i of π, for $1 \le i \le m$.

For example, $m=6$, $n=5$, and P is defined as follows: π satisfies P iff for any $i \le j$, $\pi(i) \le \pi(j)$; i.e., π is monotonic nondecreasing, and such a condition

* Dedicated to Professor L. C. Hsu, on the occasion of his 70-th Birthday.

P is called monotonic condition. Thus, if P only allows number 2 to repeat in our permutations, then $S(6,5;P)=\{122345\}$. Let π' be the only sequence 122345 in $S(6,5;P)$. Take

$$\pi=243125\in S(6,5);\ \pi_1:=\pi(4)\pi(5)=12;\ \pi_2:=\pi(1)\pi(3)=23;$$
$$\pi_3:=\pi(2)\pi(6)=45.$$

Thus, each π_i is a subsequence of π, and consider π and π_i, $1\leq i\leq 3$ as the sets of their terms, we have

$$\pi=\bigcup_{1\leq i\leq 3}\pi_i,\quad \pi_i\cap\pi_j=\varnothing,\quad \text{and}$$
$$\pi'=\pi_1\circ\pi_2\circ\pi_3=\pi(4)\pi(5)\pi(1)\pi(3)\pi(2)\pi(6)=122345.$$

Note Although both π_1 and π_2 have a number 2, these two 2's are different terms in π, namely $\pi(5)$ and $\pi(1)$. Because of this, we have $\pi_1\cap\pi_2=\varnothing$.

In the past 30 years, the problem of minimal partition of random permutations has been studied by many authors. For example, Guofeng Li[1] and the mathematicians in Science Academy of Sinica [2] considered the case that P is a monotonic nondecreasing restriction, which is so called ordered condition. In the spirit of generalizing the ordered condition, Yongjin Zhu and Ruopeng Zhu studied the problem under the quasi-ordered condition [3], and Kequan Ding worked on the pseudo-ordered case [4]. On the other hand, people tried the case with number-grouped condition, i.e., in π', all the terms of the same value must be put together. For example, π' could be 442311 or 211344 etc. In comparison with the ordered condition and its generalizations, this condition leads to a very complicated situation. Therefore, in [2], the formulation of an algorithm for minimal number-grouped partitions was annonnced as an open problem.For this problem, there have been a variety of approaches by now. For example, Jiyong Liu tried by dynamic programming [5] and Guozhi Xu, Qinghua Chen and Jiyong Liu considered the approximation of the solutions. The first algorithm for this problem, using reasonable amount of resourses, was obtained by Kequan Ding [5] in 1986, which solved the open problem.

Compare the researches in both direction, one can find that Li's algorithm is a basic subroutine for the optimization. It was proved in [8] that a genereralization of this algorithm can be used to find minimal conforming partitions for any labelled finite posets. Keep this in mind, we find that whenever we use Li's algorithm, as our subroutine, we are dealing with a monotone reconstruction problem as Li did (relabelling the terms of the sequences at hand if necessary). Here, a natural question arises:what is the nature of the monotone reconstruction problem?

Fortunataly, for this problem, we have

Theorem 1 Let A be any given sorting algorithm of (m,n) random permutations.

There is an algorithm B for the monotone reconstruction problem with at most $O(m)$ time and $O(m)$ space cost more than that used in A, and vice versa.

As its important consequence, we have the following theorem due to Zhili Wang.

Theorem 2 The time complexity of the monotone reconstruction problem for n-element random permutations is $O(n\log n)$.

Here, we are going to prove the theorems in two steps. First of all, in section 2, we consider the simplified case of our theorem 1, on $S(n,n)$. Then, in section 3, we generalize the result in section 2 to the forms mentioned above.

2 A Simplified Case $S(n,n)$

In this section, we prove a spacial version of Theorem 1, and later, in section 3, we shall use this statement to prove our Theorem 1.

Theorem 3 Let A be any given sorting algorithm of (n,n) random pemutations There is an algorithm B for the monotone reconstruction problem with at most $O(n)$ time and $O(n)$ space cost more than that used in A, and vice versa.

Proof By the definition of monotone reconstruction problem, we know that whenever we have an algorithm for this problem, we can get a minimal partition of any given random permutation. If we link the parts of this partition together such that the first term of the i-th part is right behind the last term of the $(i-1)$-th part, for each $i\geq 2$, then our permutation is well-sorted. For this reason, the substantial part of our proof is in the opposit direction, as follows.

Let A be any sorting algorithm of (n,n) random permutations. Let π be any (n,n) random permutation. Without running the risk of confusion, we use the same letter A to denote the map induced by A on $S(n,n)$. Define

$$\sigma = A(\pi) = 1\ 2\ 3\ \cdots n.$$

Suppose $j=\pi(i_j)$, $1\leq j\leq n$, we can characterize the map A by the following two-row array

$$T(\pi,\sigma)=\begin{bmatrix}1 & 2 & 3 & \cdots & n\\ i_1 & i_2 & i_3 & \cdots & i_n\end{bmatrix}$$

For example,

$$\pi = 2\ 6\ 4\ 3\ 7\ 6\ 1, \qquad \sigma = A(\pi) = 1\ 2\ 3\ 4\ 5\ 6\ 7,$$

$$T(\pi,\sigma)=\begin{bmatrix}1&2&3&4&5&6&7\\7&1&4&3&6&3&5\end{bmatrix},$$

Let $T(\pi,\sigma,i)$ be the i-th row of $T(\pi,\sigma)$, $i=1,2$. We use the following algorithm to decomposite $T(\pi,\sigma)$.

Algorithm K

a) consider $T(\pi,\sigma,2)$ as a sequence juxtaposited by some maximal increasing subsequences, i.e.,

$$T(\pi, \sigma, 2) = \tau_1 \circ \tau_2 \circ \cdots \circ \tau_k ,$$

where each τ_i is an increasing subsequence of $T(\pi, \sigma, 2)$, $1 \le i \le k$ such that $\tau_i \tau_{i+1}$ (1) is not increasing, $1 \le i \le k-1$.

b) find the partition $\{\eta_i\}_{1 \le i \le k}$ of $T(\pi, \sigma, 1)$ such that

$$T(\pi, \sigma, 1) = \eta_1 \circ \eta_2 \circ \cdots \circ \eta_k$$

and

$$|\eta_i| = |\tau_i|, \quad \text{for all } i .$$

In the example above,

$$T(\pi, \sigma, 2) = 7\ 1\ 4\ 3\ 6\ 2\ 5 ,$$

$$\tau_1 = 7, \quad \tau_2 = 14, \quad \tau_3 = 36, \quad \tau_4 = 25 .$$

$$T(\pi, \sigma, 2) = \tau_1 \circ \tau_2 \circ \tau_3 \circ \tau_4 .$$

Look at the first row $T(\pi, \sigma, 1)$ of $T(\pi, \sigma)$, the partition $\{\tau_i\}_{1 \le i \le 4}$ of $T(\pi, \sigma, 2)$ induces a partition $\{\eta_i\}_{1 \le i \le 4}$ of $T(\pi, \sigma, 1)$

$$T(\pi, \sigma, 1) = \eta_1 \circ \eta_2 \circ \eta_3 \circ \eta_4 = 1 \circ 2\ 3 \circ 4\ 5 \circ 6\ 7 ,$$

such that $|\eta_i| = |\tau_i|$, for all i.

Claim A The partition $\{\eta_i\}_{1 \le i \le k}$ is optimal; further, this is the scheme-moving-down parition of π according to $\sigma = 1\ 2\ 3 \cdots n$.

Look at the example above,

$$\pi = 2\ 6\ 4\ 3\ 7\ 5\ 1 , \qquad \sigma = 1\ 2\ 3\ 4\ 5\ 6\ 7.$$

Obviously, $\{\eta_i\}_{1 \le i \le 4}$ is the scheme-moving-down partition of π according to σ

Now, we prove the claim by induction on n. If $n = 1$, that is trivial. Suppose that is true for $n \le k$. Consider the case of $n = k+1$. Assume that $\pi(h) = n$. Then,

$$T(\pi, \sigma) = \begin{bmatrix} 1 & 2 & 3 & \cdots & n-1 & n \\ i_1 & i_2 & i_3 & \cdots & i_{n-1} & h \end{bmatrix}$$

Let

$$\pi' = \pi(1)\pi(2)\cdots\pi(h-1)\pi(h+1)\cdots\pi(n) = \pi'(1)\pi'(2)\cdots\pi'(n-1),$$

$$\sigma' = 1\ 2\ 3 \cdots n-1.$$

Then,

$$T(\pi', \sigma') = \begin{bmatrix} 1 & 2 & 3 & \cdots & n-1 \\ i_1 & i_2 & i_3 & \cdots & i_{n-1} \end{bmatrix}$$

By the induction hypothesis, we know that

$$T(\pi', \sigma', 2) = \tau_1 \circ \tau_2 \circ \cdots \circ \tau_s ,$$

where each τ_i is an increasing subsequence of $T(\pi', \sigma', 2)$, each $\tau_i \circ \tau_{i+1}$ (1) is not increasing, and

$$T(\pi', \sigma', 1) = \eta_1 \circ \eta_2 \circ \cdots \circ \eta_s .$$

such that $\{\eta_i\}_{1 \le i \le s}$ is the scheme-moving-down partition of π' according to σ'. Compare i_{n-1} and h, there are two different cases.

Case 1 $i_{n-1} \leq h$. Then $\tau'_s = \tau_s \circ \{\pi(h)\}$ is increasing. Thus, in the scheme-moving-down partition $\{\eta_i\}_{1 \leq i \leq s}$ of π', there is a part, say η_{i_0}, with its last term $(n-1)$, and in π, this $n-1$ is in front of n. Therefore, in the scheme-moving-down partition of π according to σ, the number n must be at the end of the same part as that of the $(n-1)$. And, the scheme-moving-down partitionof π according to σ is

$$\{\eta_1, \eta_2, \cdots, \eta_{i0-1}, \eta_{i0} \circ \{n\}, \eta_{i0+1}, \cdots, \eta_s\}.$$

On the other hand, since $i_{n-1} < h$, $T(\pi, \sigma, 2)$ has a partition

$$T(\pi, \sigma, 2) = \tau_1 \circ \tau_2 \circ \cdots \circ \tau_{s-1} \circ \tau''_s.$$

The corresponding partition of $T(\pi, \sigma, 1)$ is the same as the scheme-moving-down partition of π' according to σ' except that the part containing $(n-1)$ has n as the last term. Therefore, the partition obtained by our algorithm is the same as the scheme-moving-down partition for $n = k+1$. Hence, Claim A is true for all natural numbers. As the estimation of the time and space complexity is obvious, the proof of Theorem 3 is completed.

3 The Proof of Theorem 1

In this section, we consider the general case; i.e., the case for $m > n$. According to the discussion in section 2, it suffices to show that from any sorting algorithm A, an algorithm B for the monotone reconstruction problem can be developed with $O(m)$ time and $O(m)$ space cost more than that used in A.

In this case, in order to tell the difference among terms of the same value, we need to introduce the following linear order among terms of π.

Suppose that $\pi(i) = i_j$, $\forall i \in [m]$. Each term in π is characterized by an ordered pair $(\pi(i), i) = (i_j, i)$. Further, we can identify π with the two row array

$$Q(\pi) = \begin{bmatrix} \pi(1) & \pi(2) & \cdots & \pi(m) \\ 1 & 2 & \cdots & m \end{bmatrix}$$

Define an order among ordered pairs in the following way,

$$(x, y) < (u, v) \text{ iff } x < u, \text{ or } x = u \ \& y < v. \qquad (3.1)$$

Clearly, for any $\pi \in S(m,n)$, $(\{\pi(i), i)\}_{1 \leq i \leq m}, \leq)$ is a linearly ordered set. Let A be any sorting algorithm, apply A on the linearly ordered set. After doing this, as defined above, another two row array is obtained in which the first row is in the natural order of integers with repetitions and the second row is a permutation of $[m]$ without repetitions. In particular, if there are several integers in in the first row which are of the same value, then their corresponding integers in the second row are in the natural order of integers; i.e., an increasing order.

For eample, let $\pi = 2\,3\,1\,2\,3\,3\,4\,2\,4 \in S(9,4)$.

$$\pi \leftrightarrow Q(\pi)$$

where

$$Q(\pi)=\begin{bmatrix}2&3&1&2&3&3&4&2&4\\1&2&3&4&5&6&7&8&9\end{bmatrix}$$

Thus,

$$A(Q(\pi))=\begin{bmatrix}1&2&2&2&3&3&3&4&4\\3&1&4&8&2&5&6&7&9\end{bmatrix}$$

Now, we need the following algorithm to rearrange the columns in $A(Q(\pi))$.

Algorithm J

a) $i\leftarrow 1$,

b) if $i=n$, stop. Obtain $J(A(Q(\pi)))$.

c) let $(i,a)^t$ be the most right column of the form $(i,x)^t$, $(i+1,b)^t$ the most left column of the form $(i+1,y)^t$, for the given i, in $A(Q(\pi))$. If $a<b$, then $i\leftarrow i+1$, repeat b);

d) if $b<a$, find the most left column of the form $(i+1,c)^t$ such that $a<c$. If such a c does not exist, $i\leftarrow i+1$, go to b);

e) if such a c exists, let $(i+1,d)^t$ be the most right column of the form $(i+1,x)^t$, move the columns from $(i+1,c)^t$ to $(i+1,d)^t$, to the left of $(i+1\ b)^t$, and keep their relative positions unchanged. $i\leftarrow i+1$, goto b);

Here, we use an example to show the execution of step e). Suppose our array is as the following

$$A(Q(\pi))=\begin{bmatrix}1&2&2&2&3&3&3&4&4\\3&1&4&8&2&5&6&7&9\end{bmatrix}$$

Since in the second row, $3>1$ & $3<4$, we need to move the third and the forth column to the left of the second column and we have

$$J(A(Q(\pi)))=\begin{bmatrix}1&2&2&2&3&3&3&4&4\\3&4&8&1&2&5&6&7&9\end{bmatrix}$$

Thus, in the second row $J_2=J_2(A(Q(\pi)))$ of $J(A(Q(\pi)))$, we can find set $\{\tau_i\}_{1\le i\le k}$ of increasing subsequences of J_2 shch that

$$J_2=\tau_1\circ\tau_2\circ\cdots\circ\tau_k$$

and $\tau_i\circ\tau_{i+1}(1)$ is not increasing, for $i=1,2,\cdots,k-1$. Let $\{\eta_i\}_{1\le i\le k}$ be the corresponding set of subsequences of the first row $J_1=J_1(A(Q(\pi)))$ of $J(A(Q(\pi)))$; i.e., $|\eta_i|=|\tau_i|$, $i=1,2,\cdots,k$ and

$$J_1=\eta_1\circ\eta_2\circ\cdots\circ\eta_k.$$

Recall that the map from $J(A(Q(\pi)))$ to $\{\eta_i\}_{1\le i\le k}$ is K. Here, define $\Phi=K\circ J\circ A\circ Q$, where "$\circ$" is the composition operation of operators.

Claim B $\{\eta_i\}_{1\le i\le k}$ is the scheme-moving-down partition of π.

Proof of Claim B We shall prove the claim by induction on $(m-n)$ and n.

If $m-n=0$, then for all n, it is the case of Theorem 3. Suppose that claim B is true for $m-n\le k$ and $n-1$. Consider the case of $m-n=k+1$ and n. Let

$$U=\{i\,|\,\pi(i)=n, i\in[m]\},\quad h=\max U,\quad l=\min U,$$

and

$$\pi'=\pi(1)\pi(2)\cdots\pi(h-1)\pi(h+1)\cdots\pi(m);$$

i.e., π' is obtained from π by deleting the term $\pi(h)$. There are two possibilities:

Case 1 U is a singleton. Thus, $\pi'\in S(m-1,n-1)$. By the hypothesis of induction, the set $\Phi(\pi')=\{\eta_i'\}_{1\le i\le k}$ of subsequences is a scheme-moving-down partition of π'. Soppose that $\pi(s)$ is the last term in the last part η_k' of $\Phi(\pi')$. If $s<h$, then $\{\eta_i'\}_{1\le i\le k-1}\cup\{\eta_k'\circ\pi(h)\}$ is the scheme-moving-down partition of π. If $s>h$, then $\{\eta_i'\}_{1\le i\le k}\cup\{\{\pi(h)\}\}$ is the scheme-moving-down partition of π. Thus, what we need to show is that

$$\Phi(\pi)=\{\eta_i'\}_{1\le i\le k-1}\cup\{\eta_k'\circ\pi(h)\},\ \text{if } s<h \tag{3.2}$$

and

$$\Phi(\pi)=\{\eta_i'\}_{1\le i\le k}\cup\{\{\pi(h)\}\},\ \text{if } s>h.$$

By the characterization of terms in π, we know that $\pi(h)$ is characterized by $(n,h)^t$ which is the last column of $Q(\pi)$. Under the order defined by (3.1), $(n,h)^t$ is the maximal element of $(\{\pi(i),i)\}_{1\le i\le m},\le)$. Since U is a singleton, the algorithm J has nothing to do with $(n,h)^t$ (see the steps c & d)). By the induction hypothesis, $\Phi(\pi')=\{\eta_i'\}_{1\le i\le k}$ is the scheme-moving-down partition of π'. Hence, the structure of algorithm K imples that (3.2) is true.

Case 2 U is not a singleton.

As shown in the argument of case 1, $(n,h)^t$ is the maximal element of $(\{\pi(i),i)\}_{1\le i\le m},\le)$. Since U is not a singleton, $\pi(s)=n$. Thus, $1\le s\le h-1$. By the induction hypothesis, $\Phi(\pi')=\{\eta_i'\}_{1\le i\le k}$ is a scheme-moving-down partition of π'. Thus, Li's algorithm implies that $\{\eta_i'\}_{1\le i\le k-1}\cup\{\eta_k'\circ\pi(h)\}$ is the scheme-moving-down partition of π if there is no term between $\pi(s)$ and $\pi(h)$ in π which is of value n, i.e.,

$$V=\{i\,|\,s<ih,\ i\in U\}=\varnothing.$$

If $V\neq\varnothing$, then $\{\eta_i'\}_{1\le i\le k-2}\cup\{\eta_{k-1}\circ\pi(h)\}\cup\{\eta_k'\}$ is the scheme-moving-down partition of π.

Now, we show that

$$\Phi(\pi)=\{\eta_i'\}_{1\le i\le k-1}\cup\{\eta_k'\circ\pi(h)\},\ \text{if } V=\varnothing,$$
$$\Phi(\pi)=\{\eta_i'\}_{1\le i\le k-2}\cup\{\eta_{k-1}'\circ\pi(h)\}\cup\{\eta_k'\},\ \text{if } V\neq\varnothing.$$

Since U is not singleton, $V=\varnothing$ implies that algorithm J has nothing to do with all the columns of the form $(n,i)^t$. Hence, all of them except $(n,h)^t$ must be at the end of η_k' and satisfy $i<h$. It follows that the execution of algorithm K on $J(A(Q(\pi)))$ just attach $(n,h)^t$ at the end of η_k'. If $V\neq\varnothing$, the execution of algorithm J divides all the pairs of the form $(n,i)^t$ in $A(Q(\pi'))$ into two groups, say

$$G_1 = (n, i_1), (n, i_2), \cdots, (n, i_u) \mid s < i_1 < i_2 < \cdots < i_u\},$$
$$G_2 = \{(n, j_1), (n, j_2), \cdots, (n, i_v) \mid j_1 < j_2 < \cdots < j_v < s\}.$$

By the structure of algorithm K, we have

$$\eta'_{k-1} = \begin{bmatrix} \cdots & n-1 & n & \cdots & n \\ \cdots & s & i_1 & \cdots & i_u \end{bmatrix}$$

$$\eta'_k = \begin{bmatrix} n & \cdots & n \\ i_1 & \cdots & i_u \end{bmatrix}$$

Hence,

$$\Phi(\pi) = K(J(A(Q(\pi)))) = \{\eta'_i\}_{1 \le i \le k-2} \cup \{\eta'_{k-1} \circ \pi(h)\} \cup \{\eta'_k\}, \text{ if } V \neq \varnothing.$$

And, our theorem 1 is proved.

Note that the time comlexity of sorting problem of n-element random permutations is $O(n\log n)$, we know that theorem 2 is a natural consequence of theorem 1.

Reference

[1] Li, Guogeng, Science and Technology of Railway, 1 (1965) 4—9.

[2] The 2nd Group of Operations Research, Institute of Mathematics, Academia Sinica, The first research on the mathematical methods in the problem of cars marshalling, Acta Mathematicae Applicatae Sinica, 1:2 (1978) 91—105.

[3] Zhu, Yongjin and Zhu, Ruopeng, Scientia Sinica A, 2 (1983) 119—125.

[4] Ding, Kequan, Chinese Journal of Operations Research, 1 (1985) 63—64.

[5] Liu, Jiyong, On the problem of number-grouped partitions, preprint.

[6] Xu, Guozhi, Chen, Qinghua & Liu, Jiyong, An approximate algorithm of optimal digit-grouped partition of sequences, preprint.

[7] Ding, Kequan, Advances in Mathematics (Beijing), 15:1 (1986) 105—106.

[8] Ding, Kequan, Journal on Numerical Methods and Computer Applications, 8:2 (1987) 115—121.

[9] Cai, Maocheng, Acta Mathematicae Applicatae Sinica, 4:2 (1981) 140—150.

[10] Knuth, D. E., The art of computer programming, vol. 111: sorting and searching, Reading, Mass., Addison-Wesley, 1973.

[11] Baase, Sara, Computer algorithms: introduction to design and analysis, Addison-wesley, 1978.

[12] Ding, Kequan, Extremal partition theory of algorithmic combinatorics and its applications, M. S. Dissrtation, Dalian Institute of Technology, Dalian, China, 1984.