Journal of Combinatorial Theory, Series A **91**, 5–14 (2000)
doi:10.1006/jcta.2000.3087, available online at http://www.idealibrary.com on IDEAL®

On a Problem in Shuffling

Noga Alon[1]

Tel Aviv University, Tel Aviv, Israel

and

Ken Berman[2] and Daniel Kleitman[2]

Department of Mathematics, Massachusetts Institute of Technology, Cambridge, MA 02139

Communicated by the Managing Editors

Received March 10, 2000

DEDICATED TO THE MEMORY OF GIAN-CARLO ROTA

Upper and lower bounds are obtained for the number of shuffles necessary to reach the "furthest" two hand deal starting from a given permutation of a deck of cards. The bounds are on the order of $(\log_2 n)/2$ and $\log \log n$, respectively.

1. INTRODUCTION

Suppose we take a deck of n cards, shuffle it in the usual way k times, and then deal the cards all out, again in the usual way, giving n/j cards to each of j hands. We consider the following question: suppose we always start with a fresh deck, with cards in a fixed order; how large must k be, for given j and n, so that every possible set of j hands can be dealt?

The special case $j=n$, in which we ask how many shuffles are needed to obtain an arbitrary permutation of the cards from an initial start, is well known [1]; the answer is $k=\lceil \log_2 n \rceil$. The problem in general seems quite difficult. We concentrate our attention on the case $j=2$, in which the cards are to be partitioned into two equal hands by the shuffle-and-deal process. In this case we find a lower bound on the order of $(\log_2 \log_2 n)/2$ and an upper bound of $(\log_2 n)/2$. We present these results in the hope of

[1] Supported in part by a US-Israel BSF grant.
[2] Supported in part by NSA grant and a US-Israel BSF grant.

0097-3165/00 $35.00

stimulating others to improve them. As far as we know this problem was suggested by Alan Schwenk.

We begin by defining the notion of usual shuffles and deals, reviewing the permutation ($j = n$) problem, and introducing a problem closely related to ours which we shall actually consider. We then present arguments, first for the upper and then for the lower bound already mentioned.

2. DEFINITIONS, THE PERMUTATION CASE, AND THE DECLINE PROBLEM

The basic *usual* shuffle that we consider consists of splitting the cards into two blocks by cutting the deck at some point and then interleaving or merging the two blocks together in any way. Thus the order of the cards within each block is unchanged by the shuffle, and any permutation of the cards consistent with the starting orders within each of the two blocks can be achieved by a shuffle. The shuffler is free to choose the cut point and to interleave the blocks as he or she sees fit.

The usual deal consists of dealing single cards to each hand in turn; thus if there are only two hands, one gets all the cards in even positions in the deck, and the other gets those in odd positions.

We call an ordered set of j disjoint hands, each of cardinality n/j, a *deal*. In the case $n = j$, a deal is just a permutation of the cards.

We can define a directed graph among permutations, with an edge directed from p_1 to p_2 if it is possible to start with the permutation p_1, perform one shuffle, and arrive at p_2. The question we have posed, in the $n = j$ case, is in these terms: what is the diameter (longest distance between two permutations) of this directed graph?

The answer to this question is well known, and we now review it. The graph is obviously symmetric among the permutations, so without loss of generality we can set the target permutation, p_f, to be $(1, 2, ..., n)$. Notice that the card j in the jth position is lower in index than the card in the $(j+1)$st position in this target permutation, for all j.

If in the permutation p the card in the jth position is instead higher than that in the $(j+1)$st position, we call this a *decline at position j*; let $d(p)$ be the total number of declines at all positions in the permutation p.

The declines in either block of a shuffle must correspond to declines that lie somewhere between the two cards of that decline in the shuffled permutation; this means that $d(p)+1$ can decline by at most a factor of 2 in one shuffle since at least $(d(p)-1)/2$ declines must be present in one block of any cut. Moreover, it is possible to merge two blocks, each with at most d declines, to get at most d declines in the resulting permutation. Thus by

cutting at an appropriate middle decline. We can reduce $d(p)$ to $d(p')$ with $d(p') = [d(p)/2]$ at each shuffle.

These two facts together mean that it takes $\lceil \log_2 (d(p)+1) \rceil$ shuffles to reach p_f from any other permutation p.

If we let p be the reverse permutation to p_f, we find that $d(p) = n-1$, which gives the diameter, D, of our directed graph:

$$D = \lceil \log_2 n \rceil.$$

We now turn to the opposite case, in which $j = 2$; we will shuffle k times and then deal out two hands. Now many different permutations will give rise to the same deal. We can imagine that each permutation p is labeled by its deal $e(p)$. The number we seek is the distance in our graph from an arbitrary initial vertex p to the label that is furthest from it.

We assume that our target deal is the one, $e(p_f)$, that comes from the permutation $(1, 2, ..., n)$, which consists of odd numbered cards in the first hand and even numbered cards in the other.

This target deal can be described by a sequence, $(1, 2, 1, 2, ..., 1, 2)$, whose jth entry represents the number of the hand containing the jth card in this deal.

The initial permutation can also be described by a sequence. It will have n entries. half 1's and half 2's, which similarly describe the locations of the odd and even indiced cards (those which are to end up in hands 1 and 2, respectively) in the initial permutation p.

Since at this stage all we care about is the parities of the cards, we can without loss of generality assume that the final permutation, which is closest to p, is actually the increasing permutation, p_f.

Our question, for given starting sequence v, then becomes the following: how can the odd integers up to n be assigned to the 1's of v, and the even integers to the 2's, so that the resulting permutation requires the fewest shuffles to become p_f. In light of our known answer to the permutation shuffling problem, the answer to this follows from the answer to the corresponding decline problem:

If we assign the odd integers to the 1*'s and the even integers to the* 2*'s of* v *to form a permutation* p, *what is the smallest number*, $d(v)$, *of declines that* p *can have*?

The solution to our original problem (how many shuffles may be required to reach our target deal?) is then the *logarithm to base* 2 *of the maximum of* $d(v)+1$ *over all length n balanced binary sequences* v.

The statement that a permutation p has $d(p)$ declines means that p defines an ordered partition of n into $d+1$ blocks, within each of which block the entries of p are increasing. Thus, for example, the permutation 53124 has two declines and defines the ordered partition 1, 1, 3.

Thus our goal, starting with a given sequence, is *to assign odd integers to its* 1*'s and even integers to its* 2*'s so as to minimize the number of increasing blocks of the resulting permutation.*

A related question, which provides some clues as to how to approach this problem, is this:

Suppose, you are given a balanced even binary sequence v*, and a given ordered partition,* q*, of its entries* (*obtained by snipping it into pieces*)*. Is there a permutation* p*, with* $v(p) = v$*, that has* q *as increasing blocks*?

This will be possible if you can

1. pair a 1 which is the first entry of some block of p with a 2 that either immediately follows it in that block or is the first entry of some other block (these will be the entries 1 and 2 of the permutation);

2. remove these two entries from their blocks and iterate these two steps (locating the entries 3, 4 then 5, 6 of the permutation, etc.) until all the entries have been removed and the permutation is completely defined.

This will be impossible if no matter how you attempt this procedure, you always reach a point (after some step 2) at which either the top two entries of each nonempty block are all 1's or the top entries are all 2's, so that no new step is possible.

Suppose we have a partition A of the entries of v into $(1, 2)$ pairs and a partition B of the entries of v into blocks. The pairs of A can be used to construct a mapping from the entries of v to the integers up to n that preserves parity that is increasing within each block of B if and only if there is no *cycle of obstruction* among the pairs. A *cycle of obstruction* is a set of pairs such that one member of the jth pair comes before a member of the $(j+1)$st pair in the same block, for each j around the cycle (that is, including $j =$ last pair, $j+1 =$ first pair). This statement is the observation that if you attempt to pull off pairs from A that are at the front of their subsequences, you can only be stuck and fail if every front member of a pair is paired with an entry that is neither front nor immediately beyond its partner. If you then start with any front element and march along the alternating path from it to its partner and then to the top element above the partner, that element's partner, and so on, you must eventually reach a cycle of obstruction. (A single pair can be a cycle of obstruction if the paired entries lie in the same block and are either not consecutive or not in the order 12.)

3. AN UPPER BOUND FOR $d(v)$

We here describe two different procedures which lead to upper bounds on the order of $2n^{1/2}$ for $d(v)$ for any sequence v.

If v has consecutive entries 1212 or 2121, then we can match the first 1 among these with the subsequent 2 in either case and otherwise use any ordered partition into increasing blocks for v that applies to the sequence obtained from v by removing this pair of entries.

In seeking a v having n entries that requires the largest number of increasing subsequences we may therefore restrict our attention to sequences lacking these configurations. In fact, by similar reasoning we need only consider v's lacking subsequences 121 or 212 and therefore lacking any isolated 1 or 2 within v other than possibly a 2 as first entry and a 1 last.

We begin by pairing successive 12 entries of v with one another and removing them from it. Thus we *reduce*, for example (1*12*221*12*2211*122*) to (1221221112); having made this reduction, we can no longer pair a 1 with a 2 that immediately follows it; each 1 must be paired with a 2 from a different block. We call the resulting sequence the *reduced sequence* of v.

We then split the entries into two equal blocks, which in this example would be (12212) and (21112). We line these up next to one another and count how many unlike pairs of entries are in the same place in the two blocks. In this case we get

$$\begin{array}{c} 12212 \\ 21112 \end{array}$$

and there are three pairs, the first three, that are distinct in the same place.

In general, we will find some positions in which the corresponding entries in the two blocks are the same and some in which they are different.

If more pairs are the same than are different we cyclically shift the second block to find the shift which maximizes the number of unlike pairs.

Had we started with the reduced sequence 122112, the arrangement and its cyclical shifts are

$$\begin{array}{ccc} 122 & 122 & 122 \\ 112_e & 12_e1 & 2_e11 \end{array}$$

and the last of these has the most unlike corresponding pairs.

Our plan is to match the (1, 2)-entry pairs in the shifted arrangement that has the largest number of such pairs to form a consecutive odd–even pair from each. Since each entry pairs as (1, 2) at least half the time among all shifts, for some shift at least half of the pairs must be of this kind.

To accomplish this we partition our original reduced sequence into three blocks, one consisting of the first half of the entries and the next ending at the initial position of the last entry of the second block with the (1, 2)-pair maximizing shift. Thus, in the example above, the reduced permutation is divided into blocks to become 122/11/2.

After this stage we ignore these (1, 2) pairs and our problem reduces to pairing entries that are exactly like their opposite numbers in the other half of the sequence. We note that these represent at most half the pairs, and moreover, being alike, they provide two identical pairing problems, each of whose size is at most a fourth of that of the original one.

We accomplish their pairing by repeating the previous step now simultaneously on the two identical blocks of like-entry pairs. That is, we divide these in half, cyclically shift the second half to maximize the number of unlike pairs, and in the process find pairings for at least half the remaining pairs, the rest forming like-entry quartets.

With the number of blocks, $f(n')+1$, given n' original elements of the reduced permutation, we get the following recursion,

$$f(n') \leqslant 2 + 2f([n'/4]),$$

which has the solution $f(n') < 2_4^{[\log 4n']+1}\,[n']^{1/2}$.

This approach can be described as splitting the live entries in half, cyclically permuting one half to get more than half of the pairs to match, and then gluing blocks together and repeating the process.

We now give a second approach to an upper bound on $f(n')$, which leads to a similar bound, but is quite different in structure. The basic idea in it is to divide the elements into blocks and match the 1's in one block with the 2's in the next, in an order preserving way, except for one block which we chop into bits. If that block has block size $n'^{1/2}$ and there are $n'^{1/2}$ blocks we again find $f(n') < 2[n']^{1/2}$.

We begin by splitting the reduced sequence into two equal subsequences. Suppose there are more 1's than 2's in the first subsequence. We then pair the 1's in the first subsequence with the 2's in the second (and at this stage we can pair any 2's in the first with 1's in the second if they are compatible with these pairs). We are left with unpaired 2's in the first subsequence and 1's in the second.

If we now split the first subsequence at any point x, we say that a split in the second subsequence at y is *compatible* to it if every paired element above x is paired with an element above y, and vice versa. It is easy to see that every split in x is compatible with some (possibly trivial) split in y, and vice versa.

Our plan is to split the two subsequences at compatible points x and y such that the number of unpaired 2's above x is the same as the number of unpaired 1's below y. If so the number of unpaired 2's below x is the same as the number of unpaired 1's above y.

Suppose, without loss of generality, that more than half of the unpaired 2's are below x. Then after these splits, we pair these with the unpaired 1's above y.

At this point we have split our sequence into four subsequences, of which one has unpaired 2's and another has unpaired 1's, the rest being entirely paired. These can be arranged in a linear order, with all pairings between adjacent subsequences and with unpaired 1's in the first subsequence only and unpaired 2's only in the last. An important feature of this step is that the number of unpaired elements has decreased by at least a factor of 2 by the pairings in this step.

The remainder of our procedure involves repeatedly splitting these subsequences by essentially the same operation. Again we can find compatible x and y in the first and last subsequences, with the same number of such above one as below the other. Again splitting these subsequences at x and y, which now induces splits in all the intermediate subsequences, and pairing the larger of those above one and below the other reduces the number of unpaired elements by at least a factor of 2.

We continue this procedure $\log_2 n^{1/2}$ times, until the number of unpaired 1's is at most $(n')^{1/2}/2$.

At this point we can split each consecutive string of unpaired 1's into a block of its own and pair these to the unpaired 2's; this can require at most an additional $n'^{1/2}$ splittings, which gives our bound.

We now verify that the resulting blocks and pairings can have no cycle of obstruction.

Imagine that we start with a long and very narrow sheet of paper; we split it in two and glue one side of one piece to the other side of the other; we repeat this procedure until we have a roughly square piece of paper. We then split one side into a sort of a fringe.

This is essentially the nature of the procedure we have outlined. Our splits break the pairings that exist into two nonrelated pieces with no pair crossing from one piece to the other, and our new pairings link one end of one with the other end of the other by noncrossing pairs. No cycle of obstruction can ever be produced by such a procedure. Since each of the final pairings involve at least one element that is alone in its block, none of these can be in a cycle of obstruction, which ends the proof.

4. A LOWER BOUND FOR $d(v)$

Suppose we have a sequence of 1's and 2's; the *excess of* 1's between two places in the sequence is the number of 1's between them minus the number of 2's.

We obtain a lower bound here inductively by producing a sequence of 1's and 2's called a *k-stop*, *denoted as* S_k, with the following properties:

1. S_k has more 1's in it than 2's.

2. S_k contains a consecutive sequence of 2's, called a *k-blocker*, of length sufficient that it is not possible to start at any $k-1$ (not necessarily distinct) places within S_k and obtain excesses of 1's whose sum is equal to the length of the the k-blocker, where the endpoint of each excess must occur before the first $k-1$ blocker after the corresponding start.

Examples $k=1$: $S_1 = 22111 = 2^2 1^3$; the 1 blocker is 22 or 2^2;

$k=2$: $S_2 = 2^2 S_1^3$; the 2 blocker is 2^4

$k=3$: $S_3 = 2^7 S_2^8$; the 3 blocker is 211.

In general, with $a(1)=b(1)=2$, $e(1)=1$, we form S_k to contain $e(k)$ more 1 than 2 as follows:

$$S_k = 2^{b(k)} S_{k-1}^{c(k)},$$

$$b(k) = a(k) - a(k-1) \quad \text{for} \quad k>1,$$

$$a(k) = (k-1)(a(k-1)+e(k-1))+2 \quad \text{for} \quad k>1,$$

$$c(k) = (b(k)+e(k))/e(k-1) \quad \text{for} \quad k>1.$$

(We assume these numbers are all chosen to be integers.)

S_k then begins with a sequence of $a(k)$ 2's and has an excess of $a(k)+e(k)$ 1's beyond these.

We now prove that there can be no way to assign consecutive integers to the 1's and 2's of the sequence $S_k 2^{e(k)}$ so that it can be partitioned into k subsequences, with an order that is increasing in each subsequence, under our condition that 1's must be odd and 2's even.

We further claim that you cannot form a pairing of S_k, or any number of consecutive copies of S_k into consecutive odd–even pairs, that is compatible with any partition of the elements into k subsequences, even if there is an arbitrarily large reservoir of 2's (but not 1's) that can be used to pair any 1's that get in the way. Furthermore this happens because it is impossible to find enough 1's to pair with the 2's of the k-blocker.

The statement is trivial for $k=1$. Suppose, in general, the sequence is partitioned into k subsequences.

The claim follows by induction if one subsequence consists entirely of 2's; in fact the 2's in that subsequence can be absorbed into the hypothecated reservoir of 2's and we may deduce by an appropriate induction statement that it is impossible to find enough 1's to pair with the 2's of any $k-1$ blocker that lies within one of the $k-1$ other subsequences.

To prove this claim, we first assume that the first subsequence begins with the k-blocker. It must then include the entire k-blocker or else it

consists entirely of 2's. But then, until the 2's of the k-blocker have all been paired, this subsequence can contribute no 1's to the pairings of the rest. In other words, this block is inert and useless, like a mere part of the reservoir of 2's, until the 2's with which it begins have all been paired.

But while this subsequence is inert for pairing purposes there are only at most $k-1$ active subsequences from which 1's can be obtained for pairing. We may deduce by the appropriate induction hypothesis that we will not be able to pair 1's in any other subsequence across any $(k-1)$blocker. We will therefore be able to extract from each of the other $k-1$ subsequences no more than the maximum excess in each between successive $(k-1)$-blockers, which is here $a(k-1)+e(k-1)$. Summing this maximum over the $k-1$ subsequences gives a total of at most $(k-1)\,a(k-1)+(k-1)$ $e(k-1))$ excess 1's in all, not enough to exhaust the 2's of the k-blocker.

Thus, in fact *the first block is always inert*, incapable of supplying 1's. If there are any 1's in it at all, its last 2 will not be pairable at all.

Since the first block is inert in supplying pairable 1's, no $(k-1)$-blocker that is intact in one subsequence can ever be exhausted. If a $(k-1)$-blocker is separated into two or more subsequences, the count of $a(k-1)+1$ excess 1's obtainable from each subsequence will still hold.

In pairing 1's and 2's, to form the desired permutation here, eventually a $(k-1)$-blocker must be reached in some subsequence. At that point that subsequence becomes incapable of supplying 1's for pairing, and so after that point no $(k-2)$-blocker can be exhausted, and so on.

The length, $L(k)$, of this construction obeys

$$L(k) = b(k) + ((L(k-1)(n(k)+e(k))/e(k-1).$$

We can get approximate solutions to these recursions as follows:

$$e(k) = (k-1)!$$

$$a(k) \approx 2(k!),$$

$$b(k) \approx 2(k-1)(k!),$$

$$c(k) \approx (2k-1)(k-1),$$

$$L(k) \approx (2k!).$$

From this we can deduce the following:

$$\log L(k) \approx (2k/e) \log (2k/e)$$

$$k \approx e \log L(k))/(2 \log \log L(k)).$$

This implies that $d(n)$ obeys

$$d(n) \geqslant (e(\log n)/(2 \log \log n)$$

so that $(\log \log n)$ is a lower bound on the number of minimal number of shuffles needed to reach every two player deal from any given permutation of the deck.

5. FURTHER COMMENTS

It is obvious that similar considerations can be applied when there are three or more hands. We leave such application as an exercise for the reader.

There is quite a large gap between the upper and the lower bounds obtained in this note. Finding or even guessing the nature of the actual number of shuffles needed here is still an open challenge. When we think about how difficult it is to obtain a lower bound, we begin to think that $\log \log n$ is the correct answer; when contemplating the upper bounds we are tempted to think that they are hard to beat.

REFERENCE

1. Problem E3143, *Amer. Math. Monthly* **93** (1986), 299; solution, *Amer. Math. Monthly* **95** (1988), 352.