

The number of times the placement algorithm ends at the top and bottom

by John Ginsburg

ABSTRACT Let n and q be positive integers, with $n \geq 3$, and let $N = nq$. As input, we are to be given an arbitrary ordered n -sequence x_1, x_2, \dots, x_n , where $1 \leq x_i \leq N$ for all i . We are to be presented with this sequence one entry at a time. As each entry is received, it must be placed into one of the positions $1, 2, \dots, n$, where it must remain. A natural way to do this, in an attempt to sort the input sequence, is as follows. For any integer $x \in \{1, \dots, N\}$ we let $s(x)$ denote the unique integer s for which $(s-1)q + 1 \leq x \leq sq$. When we receive the entry x_i we consider those positions still unoccupied after having placed the previous $i-1$ entries, and we place x_i into the one which is closest to $s(x_i)$. In the event of a tie for closest, we choose the higher of the two positions. We refer to this procedure as *the placement algorithm*. Regarding this algorithm, we consider the following question: for how many input sequences will the last two positions filled be positions 1 and n ? We show that this number is $(n-1)^{n-3}n^2q^n$.

Key words and phrases: placement algorithm, permutation, sorting, recurrence, convolution, Abel's binomial theorem

AMS subject classification: 05A05, 05A10

Let n be a positive integer with $n \geq 3$. Consider the following problem, a very restrictive form of sorting:

We are to be presented with a random sequence of n integers x_1, x_2, \dots, x_n , one entry at a time. As each entry is received, it must be placed into one of the positions $1, 2, \dots, n$. Once placed, an entry cannot be moved; we must place it and leave it. We would like to do this in a way which will (as much as possible) sort the original sequence into correct order.

In the version of this problem we wish to consider, we are also given an integer $N \geq n$ and we are given that each of the integers to be presented is between 1 and N . To simplify our discussion, we will further assume that n divides N . Thus we suppose that $N = nq$ for some positive integer q . Statistically, we can regard our input as the result of successive sampling, n times, with repetition, from the uniformly distributed set of integers $\{1, 2, \dots, N\}$. With this in mind, we can describe a most natural algorithm. We will make use of the following definition.

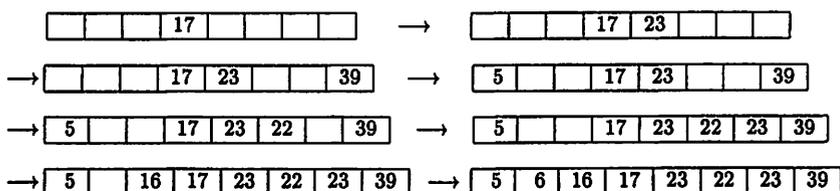
For any integer $x \in \{1, \dots, N\}$ we let $s(x)$ denote the unique integer s for which $(s - 1)q + 1 \leq x \leq sq$.

The algorithm, which we will refer to as **the placement algorithm**, is described as follows:

When we receive the entry x_i we consider those positions in the set $\{1, 2, \dots, n\}$ which are still unoccupied after having placed the previous $i - 1$ entries, and we place x_i into the one which is closest to $s(x_i)$.

In the event of a tie for closest, we choose the higher of the two positions.

Let us illustrate the placement algorithm with $n = 8$ and $q = 5$. We are to be given 8 numbers from the set $\{1, 2, \dots, 40\}$, one at a time, and we will place each one into one of the eight positions in a horizontal table, where positions 1 to 8 go from left to right. Suppose we are given the sequence 17, 23, 39, 5, 22, 23, 16, 6. The successive placements are shown below. Since $s(17) = 4$, we place the first entry 17 in position 4. Since $s(23) = 5$, and position 5 is available, we place the second entry 23 into position 5. Similarly, the third entry 39 is placed in position 8, and the fourth entry 5 in position 1. For the fifth entry, we have $s(22) = 5$. Position 5 is already occupied. The closest unoccupied position to position 5 is position 6, where we place 22. For the sixth entry, we have $s(23) = 5$. Position 5 is occupied. The closest available positions are positions 3 and 7, both two units away. We select the larger of the two and place 23 in position 7. Next, we have $s(16) = 4$. The closest unoccupied position to position 4 is position 3, where we place 16. Finally, we place the last entry in the only available position.



Many aspects of this procedure are of interest. For instance, to what extent, on average, is the output sequence better sorted than the input? We will not pursue this type of question here. Instead, we will look at a combinatorial question, relating to how the procedure ends. A second example will introduce us to the question. Again, we let $n = 8$ and $q = 5$. Suppose we apply the algorithm to the sequence 35, 9, 28, 6, 27, 11, 32, 15.

The first six entries are then placed as follows:

	9	6	11	27	28	35	
--	---	---	----	----	----	----	--

This leaves the first and last positions for the final two entries, where we would place 32(last) and 15(first). It is obvious that, no matter what the last two entries x_7 and x_8 are, when the algorithm is applied to the sequence 35, 9, 28, 6, 27, 11, x_7, x_8 it will leave the first and last positions for the final two entries. This is a rather pleasing way for the procedure to end. How often might this happen? In other words, given n and q , for how many input sequences x_1, x_2, \dots, x_n will the last two positions filled by the procedure be positions 1 and n ? We will answer this question below. It is somewhat surprising that we can find a simple, exact formula for this number.

In the sequel, any ordered sequence will be referred to as a permutation; the various entries need not be distinct. A permutation p of length n will be denoted variously by x_1, x_2, \dots, x_n or $x_1x_2\dots x_n$ or occasionally within square brackets $[x_1, x_2, \dots, x_n]$. We may also refer to the various entries of p by writing $p(i) = x_i$.

If $p = x_1, x_2, \dots, x_n$ then, for any entry x , we will let px denote the $n + 1$ -permutation x_1, x_2, \dots, x_n, x .

Let $p = x_1, x_2, \dots, x_n$. Suppose $S \subseteq \{1, 2, \dots, n\}$ with $|S| = k$ and that $S = \{i_1, i_2, \dots, i_k\}$ where $i_1 < i_2 < \dots < i_k$. We will let $p|S$ denote the k -permutation $x_{i_1}, x_{i_2}, \dots, x_{i_k}$. We refer to $p|S$ as a *subsequence* of p .

Let n be a positive integer and let k and l be positive integers such that $k + l = n$. Let p_1 be a k -permutation and let p_2 be an l -permutation. Let S be a subset of the set $\{1, 2, \dots, n\}$ such that $|S| = k$, and let $T = \{1, 2, \dots, n\} - S$. We will let $p_1 \star_S p_2$ denote the unique n -permutation p for which $p|S = p_1$ and $p|T = p_2$.

Suppose $p = x_1, x_2, \dots, x_n$ where each x_i is an integer. For any integer k , we will let $p + k$ denote the n -permutation $x_1 + k, x_2 + k, \dots, x_n + k$. Similarly we define $p - k$.

We will also refer to the placement algorithm by the letter A . When this algorithm is applied to an n -permutation $p = x_1x_2\dots x_n$ (whose entries are elements of the set $\{1, 2, \dots, nq\}$), we will let Ap denote the resulting n -permutation, which is, of course, some re-ordering of the entries of p . In the first example above, we have

$$A[17, 23, 39, 5, 22, 23, 16, 6] = [5, 6, 16, 17, 23, 22, 23, 39].$$

Given n and q , we let P_n denote the set of all n -permutations $p = x_1x_2\dots x_n$ such that $1 \leq x_i \leq nq$ for all $i = 1, 2, \dots, n$. (This set clearly depends on q as well as n , but we can think of q as fixed and n as arbitrary.) We also let

$E_n = \{p \in P_n \mid \text{the last two positions filled when } A \text{ is applied to } p \text{ are positions } 1 \text{ and } n\}$.

Thus, if $p \in E_n$, with $p = x_1x_2\dots x_n$, then either the first entry of Ap is x_{n-1} and the n 'th entry of Ap is x_n , or the first entry of Ap is x_n and the n 'th entry of Ap is x_{n-1} . Our main intention is to find a formula for $|E_n|$. As remarked above, whether or not p belongs to E_n depends only on the first $n - 2$ entries of p . For this reason, we make the following definition:

$F_n = \{x_1x_2\dots x_{n-2} \mid 1 \leq x_i \leq nq \text{ for all } i = 1, 2, \dots, n - 2 \text{ and } x_1x_2\dots x_{n-2}xy \in E_n \text{ for all } x \text{ and } y \text{ in the set } \{1, 2, \dots, nq\}\}$.

It is clear that $|E_n| = |F_n|(nq)^2$. Instead of working with E_n directly, we will prefer to work with F_n . We will use the following two items of notation. We define

$$\alpha_n = |F_n| \quad \text{and} \quad \beta_n = \frac{|F_n|}{q^{n-2}}$$

Recall that we are assuming that $n \geq 3$. It will be convenient to also set values for α_n and β_n when $n = 2$. We assign a value of 1 to each of these. That is, we set $\alpha_2 = 1$ and $\beta_2 = 1$.

We will now obtain a convolution-type recurrence relation for α_n .

Lemma 1 For all $n \geq 3$ we have

$$\alpha_n = \sum_{k=2}^{n-1} \binom{n-3}{k-2} \alpha_k \alpha_{n-k+1} (\lceil (n+k)/2 \rceil - \lceil (k+1)/2 \rceil) q$$

Proof: We remark that, for $n = 3$, we need to employ the convention that $\binom{0}{0} = 1$. The formula then states that $\alpha_3 = \binom{0}{0} \alpha_2 \alpha_2 (\lceil 5/2 \rceil - \lceil 3/2 \rceil) q$. Since $\alpha_2 = 1$, this says that $\alpha_3 = 1 \cdot 1 \cdot 1 \cdot (3 - 2) \cdot q = q$, a fact which is easy to verify directly. Thus we assume that $n \geq 4$.

Let $p = x_1x_2\dots x_{n-2}$ be any element of F_n . When algorithm A is applied to $x_1x_2\dots x_{n-2}xy$, for any x and y in $\{1, 2, \dots, nq\}$, the last two positions filled are positions 1 and n . Let k be the position into which the entry x_{n-2} is placed. Then $2 \leq k \leq n - 1$. Let $S = \{i \mid i < n - 2 \text{ and } A \text{ places } x_i \text{ into a position below position } k\}$ and let $T = \{i \mid i < n - 2 \text{ and } A \text{ places } x_i \text{ into a position above position } k\}$. Note that $S \cup T = \{1, 2, \dots, n - 3\}$. We observe that $s(x_i) < k$ for all $i \in S$. This follows from the fact that

A placed x_i below position k , at a point in the procedure when position k was available. If it were the case that $s(x_i) \geq k$ then position k would have been closer to $s(x_i)$ than the position where x_i was actually placed. This is contrary to the way the algorithm proceeds. In the same way, we see that $s(x_i) > k$ for all $i \in T$. We also note that $|S| = k - 2$ and $|T| = n - k - 1$ since all positions other than $1, n$ and k are filled by the entries x_1, x_2, \dots, x_{n-3} . Let $S = \{i_1, i_2, \dots, i_{k-2}\}$ where $i_1 < i_2 < \dots < i_{k-2}$, and let $p^- = p|_S = x_{i_1} x_{i_2} \dots x_{i_{k-2}}$. Since $s(x_i) < k$ for all $i \in S$, we have $x_i \leq (k - 1)q$ for all $i \in S$. We claim that $p^- \in F_k$. We can see this as follows. When the algorithm is applied to p^- , on $\{1, 2, \dots, k\}$, each x_{i_j} is successively assigned the position closest to $s(x_{i_j})$ among $1, 2, \dots, k$ which hasn't been assigned to any x_{i_r} with $r < j$. When x_{i_j} was assigned a position when the algorithm was applied to p on $\{1, 2, \dots, n\}$, all the entries x_{i_r} with $r < j$ had been assigned positions, and the position closest to $s(x_{i_j})$ which was available was one below k . Any entries x_i , for $i \in T$, which had been placed before this point, were placed in positions above k . It follows that the positions assigned to the entries x_{i_j} when A is applied to p^- , on $\{1, 2, \dots, k\}$ are identical to those assigned to those entries when A is applied to p on $\{1, 2, \dots, n\}$. After these entries are placed, the two positions among $1, 2, \dots, k$ which have not been filled are positions 1 and k . This means that $p^- \in F_k$. We can apply the same reasoning to the permutation $p|_T$, but we need to first reduce its entries. Since $s(x_i) > k$ for all $i \in T$, we have $x_i \geq kq + 1$ for all $i \in T$. Now we let $p^+ = (p|_T) - (k - 1)q$. Then p^+ is a permutation of elements from the set $\{1, 2, \dots, (n - k + 1)q\}$, of length $n - k - 1$. Now, apply the algorithm to the entries of p^+ on the set $\{1, 2, \dots, n - k + 1\}$. If we add $k - 1$ to each of the positions assigned, we get exactly the positions assigned to the corresponding entries of $p|_T$ when A was applied to p on $\{1, 2, \dots, n\}$. Thus the two positions among $\{1, 2, \dots, n - k + 1\}$ which are not filled when the algorithm is applied to p^+ on the set $\{1, 2, \dots, (n - k + 1)q\}$ are positions $n - (k - 1) = n - k + 1$ and $k - (k - 1) = 1$. Therefore $p^+ \in F_{n-k+1}$. Our definitions imply that $p = (p^- *_{S} (p^+ + (k - 1)q)) x_{n-2}$.

Now let us see how we can reverse this reasoning. Let k be any integer such that $2 \leq k \leq n - 1$. Let x be any element of the set $\{1, 2, \dots, nq\}$ such that $s(x)$ is closer to k than it is to n , and also such that $s(x)$ is at least as close to k as it is to 1 . (remember that the algorithm chooses upwards in case of a tie). Also let $p^- \in F_k$, let $p^+ \in F_{n-k+1}$ and let S be any subset of the set $\{1, 2, \dots, n - 3\}$ with $|S| = k - 2$. We define $p = (p^- *_{S} (p^+ + (k - 1)q)) x$. Then p is an $n - 2$ -permutation of the set $\{1, 2, \dots, nq\}$. We claim that $p \in F_n$. To see this, imagine that we append any two elements from the set $\{1, 2, \dots, nq\}$ to the end of p and then apply the procedure A , assigning positions in $\{1, 2, \dots, n\}$. The entries belonging

to p^- will be assigned exactly the same positions as they would when A is applied to p^- by itself, assigning positions in the set $\{1, 2, \dots, k\}$: they are assigned positions in exactly the same order, and even though other positions (above k) may now be available in the former, none of these can be preferable to the positions assigned in the latter, because this would mean that position k itself was preferable in the latter. But this cannot be the case, since position k was never assigned in the procedure on $\{1, 2, \dots, k\}$, owing to the fact that $p^- \in F_k$. Similarly, since position 1 is never assigned to any of the entries of p^+ , it follows that the positions assigned to the entries of $p^+ + (k-1)q$ are completely unaffected by the possible availability of positions below k ; the algorithm places them just as it does when applied to p^+ by itself, (translated upwards by $k-1$), assigning positions in the set $\{k, k+1, \dots, n\}$. Therefore, once the algorithm has placed all the entries of $p^- \star_S (p^+ + (k-1)q)$, the positions which remain unassigned are positions 1, k and n . Next comes x , and because of the way it was chosen, it will be placed in position k . This leaves exactly positions 1 and n for the last two entries. It follows that $p \in F_n$.

Let's now interpret, in a direct way, the conditions on x in the preceding argument. To say that $s(x)$ is at least as close to k as it is to 1 is equivalent to saying that $s(x) \geq \lceil (k+1)/2 \rceil$. And to say that $s(x)$ is closer to k than it is to n is equivalent to saying that $s(x) \leq \lfloor (n+k)/2 \rfloor - 1$. Therefore, the integers x which can serve to construct p in the preceding argument are exactly the ones for which $\lceil (k+1)/2 \rceil \leq s(x) \leq \lfloor (n+k)/2 \rfloor - 1$, or equivalently, for which

$$(\lceil (k+1)/2 \rceil - 1)q + 1 \leq x \leq (\lfloor (n+k)/2 \rfloor - 1)q.$$

The number of such integers x is thus

$$(\lfloor (n+k)/2 \rfloor - 1)q - (\lceil (k+1)/2 \rceil - 1)q = (\lfloor (n+k)/2 \rfloor - \lceil (k+1)/2 \rceil)q.$$

Let us combine this fact with the preceding argument. For any given integer k for which $2 \leq k \leq n-1$, the number of quadruples (p^-, S, p^+, x) such that $p^- \in F_k$, S is a subset of the set $\{1, 2, \dots, n-3\}$ with $|S| = k-2$, $p^+ \in F_{n-k+1}$ and $\lceil (k+1)/2 \rceil \leq s(x) \leq \lfloor (n+k)/2 \rfloor - 1$ is

$$\begin{aligned} & |F_k| \binom{n-3}{k-2} |F_{n-k+1}| (\lfloor (n+k)/2 \rfloor - \lceil (k+1)/2 \rceil) q \\ &= \binom{n-3}{k-2} \alpha_k \alpha_{n-k+1} (\lfloor (n+k)/2 \rfloor - \lceil (k+1)/2 \rceil) q. \end{aligned}$$

Each such quadruple gives rise to an element $p = (p^- \star_S (p^+ + (k-1)q)) x$ of F_n . Since we can identify within this expression the entries of p which are assigned positions below (and above) position k , it is easy to see that the association of (p^-, S, p^+, x) to p is one-to-one. Furthermore, if $k_1 \neq k_2$, the set of permutations p constructed from $k = k_1$ in this way is disjoint

from the set of permutations p constructed from $k = k_2$. (The algorithm places the $n - 2$ 'th entry of p into position k_1 for each p in the former, and into position k_2 for each p in the latter.) The first part of the proof shows that the union of these sets, over all such k , includes all of the elements of F_n . Summing over k concludes the proof. \circ

The difference of ceilings in our recurrence relation at first seems to be somewhat disheartening. With a bit of patience, and a bit of work on the form of the summation, we will be able to solve the recurrence, using one of Abel's binomial formulas to ride in to the rescue. To begin with, by considering the possible parities of n and k , we obtain the following helpful fact:

$$\lceil (n+k)/2 \rceil - \lceil (k+1)/2 \rceil = \begin{cases} \lfloor n/2 \rfloor & \text{if } n \text{ is odd or } k \text{ is odd} \\ \lfloor n/2 \rfloor - 1 & \text{if } n \text{ and } k \text{ are both even} \end{cases}$$

Now, let's rewrite the recurrence in Lemma 1 in terms of $\beta_n = \alpha_n/q^{n-2}$, in the process eliminating the q factor. Multiply both sides of the recurrence by $1/q^{n-2}$. When we take the multiplier inside the summation, split it into $1/q^{k-2} \cdot 1/q^{n-k-1} \cdot 1/q$. We get the following.

$$\beta_n = \sum_{k=2}^{n-1} \binom{n-3}{k-2} \beta_k \beta_{n-k+1} (\lceil (n+k)/2 \rceil - \lceil (k+1)/2 \rceil) \quad \text{for } n \geq 3$$

Next we rewrite this last recurrence relation in a more standard form. Define a new sequence of numbers γ_n , for $n = 0, 1, 2, \dots$ by setting $\gamma_n = \beta_{n+2}$. The preceding formula becomes

$$\gamma_{n-2} = \sum_{k=2}^{n-1} \binom{n-3}{k-2} \gamma_{k-2} \gamma_{n-k-1} (\lceil (n+k)/2 \rceil - \lceil (k+1)/2 \rceil) \quad \text{for } n \geq 3$$

Next, let $l = k - 2$ and re-index the sum to get

$$\gamma_{n-2} = \sum_{l=0}^{n-3} \binom{n-3}{l} \gamma_l \gamma_{n-l-3} (\lceil (n+l+2)/2 \rceil - \lceil (l+3)/2 \rceil) \quad \text{for } n \geq 3$$

Now, let $m = n - 3$ to get

$$\gamma_{m+1} = \sum_{l=0}^m \binom{m}{l} \gamma_l \gamma_{m-l} (\lceil (m+l+5)/2 \rceil - \lceil (l+3)/2 \rceil) \quad \text{for } m \geq 0$$

Finally, restore the original symbols n and k in place of m and l , to get the

following version, which is the one we will use.

Lemma 2 For $n \geq 0$ we have

$$\gamma_{n+1} = \sum_{k=0}^n \binom{n}{k} \gamma_k \gamma_{n-k} (\lceil (n+k+5)/2 \rceil - \lceil (k+3)/2 \rceil)$$

We note that $\gamma_0 = \beta_2 = 1$. Taking into account the new form of the difference of ceilings in Lemma 2, we rewrite our previous formula as follows:

$$\lceil (n+k+5)/2 \rceil - \lceil (k+3)/2 \rceil = \begin{cases} \lfloor (n+3)/2 \rfloor & \text{if } n \text{ is even or } k \text{ is odd} \\ \lfloor (n+3)/2 \rfloor - 1 & \text{if } n \text{ is odd and } k \text{ is even} \end{cases}$$

We will refer to this below as our *parity formula*.

We now recall one of Abel's binomial formulas. There are various versions of these formulas, broad generalizations of the Binomial Theorem. As references for these formulas, we recommend [2] pages 19, 173-174, [3] pages 18-23, [1] and [4]. The formula we will employ, valid for any nonnegative integer n , is the following. (for example, see 44(b) on page 19 of [2]).

Abel's binomial formula

$$\sum_{k=0}^n \binom{n}{k} (x+k)^{k-1} (y+(n-k))^{n-k-1} = \left(\frac{1}{x} + \frac{1}{y} \right) (x+y+n)^{n-1}.$$

Taking $x = 1$ and $y = 1$ gives the particular case we will need:

$$\sum_{k=0}^n \binom{n}{k} (k+1)^{k-1} (n-k+1)^{n-k-1} = 2(n+2)^{n-1}.$$

We will now show that the formula $\gamma_n = (n+1)^{n-1}$ is a solution for the recurrence relation in Lemma 2. Since this formula also gives the value 1 when $n = 0$, it must be the unique solution of the recurrence. Thus what we need to show is the following.

Lemma 3

$$(n+2)^n = \sum_{k=0}^n \binom{n}{k} (k+1)^{k-1} (n-k+1)^{n-k-1} (\lceil (n+k+5)/2 \rceil - \lceil (k+3)/2 \rceil).$$

Proof: Consider first the case when n is even. Our parity formula above implies that, in this case, $\lceil (n+k+5)/2 \rceil - \lceil (k+3)/2 \rceil = \lfloor (n+3)/2 \rfloor = (n+2)/2$. The formula we wish to establish becomes

$$(n+2)^n = \sum_{k=0}^n \binom{n}{k} (k+1)^{k-1} (n-k+1)^{n-k-1} (n+2)/2, \text{ or, in other words}$$

$$(n+2)^n = (n+2)/2 \sum_{k=0}^n \binom{n}{k} (k+1)^{k-1} (n-k+1)^{n-k-1}.$$

This follows immediately from Abel's formula above.

Next consider the case when n is odd. We split the sum into two parts and again use the above parity formula:

$$\begin{aligned} & \sum_{k=0}^n \binom{n}{k} (k+1)^{k-1} (n-k+1)^{n-k-1} (\lceil (n+k+5)/2 \rceil - \lceil (k+3)/2 \rceil) \\ &= \sum_{k \text{ even}} \binom{n}{k} (k+1)^{k-1} (n-k+1)^{n-k-1} (\lceil (n+k+5)/2 \rceil - \lceil (k+3)/2 \rceil) \\ &+ \sum_{k \text{ odd}} \binom{n}{k} (k+1)^{k-1} (n-k+1)^{n-k-1} (\lceil (n+k+5)/2 \rceil - \lceil (k+3)/2 \rceil) \\ &= \sum_{k \text{ even}} \binom{n}{k} (k+1)^{k-1} (n-k+1)^{n-k-1} (\lfloor (n+3)/2 \rfloor - 1) \\ &+ \sum_{k \text{ odd}} \binom{n}{k} (k+1)^{k-1} (n-k+1)^{n-k-1} \lfloor (n+3)/2 \rfloor \\ &= \sum_{k \text{ even}} \binom{n}{k} (k+1)^{k-1} (n-k+1)^{n-k-1} ((n+3)/2 - 1) \\ &+ \sum_{k \text{ odd}} \binom{n}{k} (k+1)^{k-1} (n-k+1)^{n-k-1} (n+3)/2 \\ &= \sum_{k=0}^n \binom{n}{k} (k+1)^{k-1} (n-k+1)^{n-k-1} (n+1)/2 \\ &+ \sum_{k \text{ odd}} \binom{n}{k} (k+1)^{k-1} (n-k+1)^{n-k-1} \\ &= (n+1)/2 \sum_{k=0}^n \binom{n}{k} (k+1)^{k-1} (n-k+1)^{n-k-1} \\ &+ \sum_{k \text{ odd}} \binom{n}{k} (k+1)^{k-1} (n-k+1)^{n-k-1} \end{aligned}$$

$$\begin{aligned}
&= (n+1)/2 (2(n+2)^{n-1}) + \sum_{k \text{ odd}} \binom{n}{k} (k+1)^{k-1} (n-k+1)^{n-k-1} \\
&= (n+1)(n+2)^{n-1} + \sum_{k \text{ odd}} \binom{n}{k} (k+1)^{k-1} (n-k+1)^{n-k-1}
\end{aligned}$$

where we have again used Abel's formula. We will now show that

$$\sum_{k \text{ odd}} \binom{n}{k} (k+1)^{k-1} (n-k+1)^{n-k-1} = (n+2)^{n-1}. \text{ This will imply that}$$

$$\begin{aligned}
&(n+1)(n+2)^{n-1} + \sum_{k \text{ odd}} \binom{n}{k} (k+1)^{k-1} (n-k+1)^{n-k-1} \\
&= (n+1)(n+2)^{n-1} + (n+2)^{n-1} \\
&= (n+2)(n+2)^{n-1} = (n+2)^n, \text{ which will complete the proof of Lemma 3.}
\end{aligned}$$

To show that $\sum_{k \text{ odd}} \binom{n}{k} (k+1)^{k-1} (n-k+1)^{n-k-1} = (n+2)^{n-1}$, we will show that

$$\sum_{k \text{ odd}} \binom{n}{k} (k+1)^{k-1} (n-k+1)^{n-k-1} = \sum_{k \text{ even}} \binom{n}{k} (k+1)^{k-1} (n-k+1)^{n-k-1}$$

Since, by Abel's formula, these last two sums together add up to $2(n+2)^{n-1}$, this gives us what we want. In the first sum, over odd k , we let $l = n - k$. Since n is odd, this changes the sum into one over all even values of l from 0 to n . This sum becomes

$$\sum_{l \text{ even}} \binom{n}{n-l} (n-l+1)^{n-l-1} (l+1)^{l-1} = \sum_{l \text{ even}} \binom{n}{l} (n-l+1)^{n-l-1} (l+1)^{l-1}$$

which is identical to the second sum. \circ

Our formula now follows directly.

Theorem Let n and q be positive integers, with $n \geq 3$. The number of n -permutations of the set $\{1, 2, \dots, nq\}$ for which the last two positions filled by the placement algorithm are the top and bottom is $(n-1)^{n-3}n^2q^n$.

Proof: We have $|E_n| = |F_n| (nq)^2 = \alpha_n(nq)^2 = \beta_n q^{n-2} (nq)^2 = \gamma_{n-2} q^{n-2} (nq)^2 = (n-1)^{n-3} q^{n-2} (nq)^2 = (n-1)^{n-3} n^2 q^n. \circ$

I am grateful to H. Wilf and to D. Zeilberger for their kind and helpful communications with me regarding the problem considered in this paper.

In Memoriam This paper is dedicated to the memory of my parents, Evelyn (Pollock) Ginsburg and Mervin Ginsburg.

References

- [1] S. Ekhad and J. Majewicz, A short WZ-style proof of Abel's identity, *Electronic Journal of Combinatorics* 3,(1996), R16.
- [2] L. Lovász, *Combinatorial Problems and Exercises*, North-Holland Publishing Company, New York, 1979.
- [3] J. Riordan, *Combinatorial Identities*, Robert E. Krieger Publishing Company, Inc. Huntington, New York, 1979.
- [4] S. Roman, *The Abel Polynomials*, 4.1.5; *The Umbral Calculus*, Academic Press, New York, 1984.

Department of Mathematics and Statistics, University of Winnipeg
Winnipeg, Manitoba, Canada(j.ginsburg@uwinnipeg.ca)