PERMUTING IN PLACE: ANALYSIS OF TWO STOPPING RULES

ALOIS PANHOLZER, HELMUT PRODINGER†, AND MARKO RIEDEL

Abstract. Permuting in place has been first analyzed by Knuth. It uses the cycle structure of the permutation. The elements of an array to be permuted are only moved when one sees a cycle leader (smallest element in its cycle). So the essential part of such an algorithm is to test an element i about whether it is a cycle leader.

Recently, Keller [3] introduced two stopping rules: "If the last cycle leader has been detected, all elements have been moved, and no further tests are necessary" (Heuristic 1), resp. "If only r elements have not been moved, then proceeding along a cycle is only useful for r steps" (Heuristic 2).

We analyze the average costs of these modifications applied to the standard algorithm of Knuth; they are $(n+2)H_n - \frac{5n}{2} - \frac{1}{2} \sim n\log n$ resp. $\frac{2n+1}{4}H_{\lfloor \frac{n+1}{2}\rfloor} + \frac{1}{2}H_{2\lfloor \frac{n+1}{2}\rfloor} - \frac{1}{2}(\lfloor \frac{n+1}{2}\rfloor - \lfloor \frac{n}{2}\rfloor) - \frac{n+1}{2} \sim \frac{n}{2}\log n$, as opposed to $(n+1)H_n - 2n \sim n\log n$ in the classical case.

1. Introduction and the classical algorithm

Permuting in place has been first analyzed by Knuth [5]; compare also [4, 6]. Further developments can be found in [1].

The algorithm is supposed to permute the elements of a one-dimensional array $(x_1, \ldots, x_n)$ in place by a permutation π, i. e., replacing the array by $(x_{\pi(1)}, \ldots, x_{\pi(n)})$, using only a bounded amount of auxiliary memory. The permutation π is not allowed to be changed. A most natural way to achieve this is to follow the cycle structure of π. The essential thing is to test an element i about whether it is a cycle leader (the smallest element in its cycle). Once a cycle leader has been found, the elements in the array with indices belonging to that cycle can be easily permuted, and one does this until all cycles have appeared. Naturally, one has to test each individual element whether it is a cycle leader. The classical cycle leader checking algorithm which has been analyzed by Knuth in [5], is given here as Algorithm 1.

The total number of steps to test all the elements $1, \ldots, n$, measured by the number of passes of the `while`-loop in Algorithm 1, is a fundamental parameter for permutation in place. Knuth called it simply "a" (a precise definition follows soon). As Knuth has demonstrated in his fundamental analysis, it is important to consider for a permutation π its canonical representation $q_1 \ldots q_n$ as a product of cycles. It is obtained by arranging the cycles with the cycle leaders in first place, and then arranging them according to their cycle leaders in decreasing order. This is also well described in the book [8].

Date: January 24, 2004.

Key words and phrases. Analysis of Algorithms, permuting in place, Knuth's parameter a, harmonic numbers.

Part of this work was done during the first author's visit to the John Knopfmacher Centre for Applicable Analysis and Number Theory at the University of the Witwatersrand, Johannesburg, South Africa.

†Supported by NRF Grant 2053748.

Algorithm 1 Classical cycle leader checking algorithm.

Input: $\pi = (\pi(1), \dots, \pi(n))$ permutation of length n, $1 \le i \le n$ integer.
Output: leader $\in \{\text{true}, \text{false}\}$, leader $=$ true iff i is a cycle leader.

```
leader ← true
j ← π(i)
while j > i do
   j ← π(j)
end while
if j ≠ i then
   leader ← false
end if
return leader
```

Knuth's "parameter a" is much easier to understand in this setting and is then for a given permutation $q_1 \dots q_n$ defined by

$$a = \#\{(i,j) \mid 1 \le i < j \le n, \quad q_i < q_{i+1}, q_i < q_{i+2}, \dots, q_i < q_j\}. \tag{1}$$

There are simple methods to go back and forth between $(\pi(1), \dots, \pi(n))$ and $q_1 \dots q_n$. For all our considerations, we assume that a permutation is given as $q_1 \dots q_n$.

In the average-case analysis of the algorithms, we will always assume, that all $n!$ permutations of length n can occur as permutations $q_1 \dots q_n$ with equal probability; then Knuth's "parameter a" is on average given by

$$\mathbb{E}(a) = \sum_{1 \le i < j \le n} \frac{1}{j - i + 1} = (n+1)H_n - 2n. \tag{2}$$

(The numbers involved are the harmonic numbers $H_n = \sum_{1 \le k \le n} 1/k$.)

In the recent paper [3], Keller introduced two stopping rules. For his experiments comparing the savings of execution times he used the number of *function calls* of `pi` (evaluations $\pi(i)$ for a given permutation π and any element i) in the cycle leader checking algorithm `IsLeader`. This parameter, which is denoted here by κ, is related to the a parameter as follows. For each element, which is tested in the algorithm `IsLeader` about being a cycle leader, Keller counts one more. (Because of the various stopping rules under consideration, not every element must be tested, therefore the difference is not n for all the rules.)

In the classical algorithm, we have of course

$$\kappa = \#\{(i,j) \mid 1 \le i \le j \le n, \quad q_i \le q_{i+1}, q_i \le q_{i+2}, \dots, q_i \le q_j\}, \tag{3}$$

thus $\kappa = a + n$ for every permutation of length n, and we have in particular

$$\mathbb{E}(\kappa) = (n+1)H_n - n. \tag{4}$$

Keller says that algorithm **A1** (the simple algorithm analyzed by Knuth) behaves on the average like $\frac{2}{3} n \log n$, which does not match with the result $(n+1)H_n - n$. However[1] Keller means the binary logarithm when he writes log, and $2/(3 \log 2) \approx 0.96$. Probably one cannot distinguish this value from the true constant 1 even with Keller's elaborate

[1] Personal communication

experiments. That is why we *have* to analyze algorithms! There are insights that only a thorough theoretical analysis can unveil.

In [5], the variance is already given:

$$\mathbb{V}(a) = \mathbb{V}(\kappa) = 2n^2 - (n+1)^2 H_n^{(2)} - (n+1)H_n + 4n, \tag{5}$$

which is asymptotically equivalent to $(2 - \frac{\pi^2}{6})n^2$. (Here the second order harmonic numbers $H_n^{(2)} = \sum_{1\le k\le n} 1/k^2$ also appear.) In [6], q-analogues for the expectation and the variance of this parameter a are given.

Knuth said in 1972 that the corresponding computation should have been done by a computer. Indeed, recently Carsten Schneider [7] has written an impressive Ph. D. thesis that allows to manipulate intricate sums involving harmonic numbers with a computer. However, his software is not yet in the public domain. For Knuth's original parameter "a" alternative methods using generating functions are available, see [4, 8]. Certain functional equations allow then, at least in principle, to compute all moments, which was demonstrated in [4]. It was shown there, that asymptotically for $n \to \infty$, the s-th moment $\mathbb{E}(a^s)$ is given by $n^s \log^s n + (\gamma - 2)sn^s \log^{s-1} n + \mathcal{O}(n^s \log^{s-2} n)$, where γ denotes the Euler constant. Studying these equations further, it is even possible to obtain limiting distribution results, as was sketched in [2].

While this sounds promising, for the various modifications (stopping rules) studied in this paper, such manageable functional equations do not seem to exist. One is thus back to the methods that Knuth used already 30 years ago!

Since no theoretical analysis of the average cost of the standard permutation algorithm (called **A1** in [3]) together with the stopping rules was provided in [3], we study these parameters a resp. κ in the sequel. The remaining algorithms (**A2** and **A3** in [3]) seem to be more complicated and are left for future research, as well as the *bit complexity* of the algorithms[2].

Since Knuth and Keller consider two quantities that differ slightly, we analyzed both of them; mathematically, both versions are equally difficult. However, for Keller's heuristic 2, there is a third variant to be analyzed, namely his implementation of this heuristic. Although this implemenation is slightly inferior to the one presented in this paper, we do so to be able to compare experimental data from Keller's paper with theoretical formulæ.

2. Analysis of Keller's heuristic 1

The first heuristic given in [3] and denoted `Fastbreak 1` uses the fact that if the last cycle leader has been detected, all elements have been moved (this can be checked easily by simply counting the number of elements already moved), and thus no further tests are necessary. This means that for the remaining elements, the cycle leader checking algorithm Algorithm 1 is not called anymore. That is included in the permutation algorithm given in [3, Fig. 3].

Intuitively, one will expect, that for a length-n permutation the number of function calls (measured by κ) will decrease by this heuristic on average by about n: a length-n permutation has on average $\log n$ cycles and the expected length of a cycle

[2]A referee suggested to study this.

is $n/\log n$; thus the expected savings due to not checking the last cycle are about $n/\log n \cdot H_{\lfloor n/\log n\rfloor} \sim n$. Our exact computations will confirm this intuition.

We assume that the elements $1, 2, \dots, n$ are tested in that order. However, the last cycle leader is the first element q_1 in the canonical representation!

We will first treat Knuth's parameter $a^{[\mathrm{FB1}]}$ and must consequently consider

$$a^{[\mathrm{FB1}]} = \#\{(i,j) \mid 1 \le i < j \le n, \quad q_i < q_{i+1}, \dots, q_i < q_j \ \textbf{and} \ q_i \le q_1\}. \tag{6}$$

It is a good idea to recall the original analysis of Knuth: He introduced indicator variables y_{ij}, defined by

$$y_{ij} = \begin{cases} 1 & \text{if } q_i = \min\{q_i, q_{i+1}, \dots, q_j\}, \\ 0 & \text{otherwise.} \end{cases}$$

Then the average of "parameter a" is given by

$$\mathbb{E}(a) = \sum_{1 \le i < j \le n} \mathbb{E}(y_{ij});$$

note that $\mathbb{E}(y_{ij})$ is just the probability that $q_i = \min\{q_i, q_{i+1}, \dots, q_j\}$.

For the modification $a^{[\mathrm{FB1}]}$ it is a good idea to split the contributions into $i = 1$ and $i \ge 2$, since

$$y_{ij}^{[\mathrm{FB1}]} = \begin{cases} 1 & \text{if } q_i = \min\{q_i, q_{i+1}, \dots, q_j\} \ \textbf{and} \ q_i \le q_1, \\ 0 & \text{otherwise.} \end{cases}$$

We start with the instance $i \ge 2$. Assume that $q_1 = L$. We compute the probability that $q_i = \min\{q_i, q_{i+1}, \dots, q_j\}$ **and** $q_i < L$. It is perhaps slightly easier to think about $q_2 \dots q_n$ as a random permutation of $\{1, \dots, n-1\}$, and a number L with $0 \le L \le n-1$ being given; we compute the probability that $q_i = \min\{q_i, q_{i+1}, \dots, q_j\}$ **and** $q_i \le L$. It is

$$\sum_{k=1}^{L} \frac{1}{n-1} \frac{\binom{n-1-k}{j-i}}{\binom{n-2}{j-i}} = \frac{1}{(n-1)\binom{n-2}{j-i}} \left[\binom{n-1}{j-i+1} - \binom{n-1-L}{j-i+1} \right].$$

We need the average of this over $L = 0, \dots, n-1$:

$$\begin{aligned} \frac{1}{n} \sum_{L=0}^{n-1} \frac{1}{(n-1)\binom{n-2}{j-i}} &\left[\binom{n-1}{j-i+1} - \binom{n-1-L}{j-i+1} \right] \\ &= \frac{1}{j-i+1} - \frac{1}{n(n-1)\binom{n-2}{j-i}} \binom{n}{j-i+2} \\ &= \frac{1}{j-i+1} - \frac{1}{(j-i+2)(j-i+1)} \\ &= \frac{1}{j-i+2}. \end{aligned}$$

For $i = 1$ we get $1/j$, as in the traditional case. Thus

$$\begin{aligned}\mathbb{E}(a^{[\mathrm{FB1}]}) = \sum_{1\le i<j\le n} \mathbb{E}(y_{ij}^{[\mathrm{FB1}]}) &= \sum_{2\le j\le n} \frac{1}{j} + \sum_{2\le i<j\le n} \frac{1}{j-i+2}\\ &= H_n - 1 + \sum_{3\le r\le n} \frac{n+1-r}{r}\\ &= H_n - 1 + (n+1)\Big(H_n - \frac{3}{2}\Big) - (n-2)\\ &= (n+2)H_n - \frac{5n}{2} - \frac{1}{2}. \end{aligned} \tag{7}$$

For $n = 1, 2, 3, 4, \ldots$ we get the values $0, \frac{1}{2}, \frac{7}{6}, 2, \ldots$, which checks. Experiments with permutations of up to 100 elements checked as well.

Knuth sketches this variation already in his original paper [5]. Comparing this result with the formula $(n+1)H_n - 2n$ for the standard case, we see that asymptotically (on average) there are no savings obtained by using this stopping rule.

The computations for $\kappa^{[\mathrm{FB1}]}$ are analogous; for completeness we give its definition

$$\kappa^{[\mathrm{FB1}]} = \#\{(i,j) \mid 1 \le i \le j \le n, \quad q_i \le q_{i+1}, \ldots, q_i \le q_j \textbf{ and } q_i \le q_1\} \tag{8}$$

and get

$$\mathbb{E}(\kappa^{[\mathrm{FB1}]}) = \sum_{1\le j\le n} \frac{1}{j} + \sum_{1\le i\le j\le n} \frac{1}{j-i+2},$$

which gives

$$\mathbb{E}(\kappa^{[\mathrm{FB1}]}) = (n+2)H_n - 2n. \tag{9}$$

Next we also sketch the computation of the variances $\mathbb{V}(a^{[\mathrm{FB1}]})$ resp. $\mathbb{V}(\kappa^{[\mathrm{FB1}]})$. The second moment $\mathbb{E}\big((a^{[\mathrm{FB1}]})^2\big)$ for Knuth's parameter a is given by

$$\mathbb{E}\big((a^{[\mathrm{FB1}]})^2\big) = \mathbb{E}(a^{[\mathrm{FB1}]}) + \sum_{\substack{1\le i_1<j_1\le n,\\ 1\le i_2<j_2\le n,\\ (i_1,j_1)\ne(i_2,j_2)}} \mathbb{E}\big(y_{i_1j_1}^{[\mathrm{FB1}]} y_{i_2j_2}^{[\mathrm{FB1}]}\big),$$

and by considering all possible cases we get

$$\begin{aligned}\mathbb{E}\big((a^{[\mathrm{FB1}]})^2\big) &= (n+2)H_n - \frac{5n}{2} - \frac{1}{2}\\ &\quad + 2\sum_{2\le i_1<j_1<i_2<j_2\le n} \frac{1}{j_1-i_1+j_2-i_2+3}\left(\frac{1}{j_2-i_2+2} + \frac{1}{j_1-i_1+2}\right)\\ &\quad + 2\sum_{2\le i_1<i_2<j_2\le j_1\le n} \frac{1}{(j_1-i_1+2)(j_2-i_2+2)}\\ &\quad + 2\sum_{2\le i_1<i_2\le j_1<j_2\le n} \frac{1}{(j_2-i_1+2)(j_2-i_2+2)} + 2\sum_{2\le i<j_1<j_2\le n} \frac{1}{j_2-i+2}\\ &\quad + 2\sum_{2\le j_1<i<j_2\le n} \frac{1}{(j_2-i+j_1+1)j_1} + 2\sum_{2\le j_1<j_2\le n} \frac{1}{j_2}. \end{aligned}$$

(The last summations cover the cases, where $i := i_1 = i_2$, resp. $i_1 = i_2 = 1$.)

As an example, we consider the range $2 \le i_1 < i_2 \le j_1 < j_2 \le n$ and describe how to compute the quantity $\mathbb{E}\big(y_{i_1j_1}^{[\mathrm{FB1}]}y_{i_2j_2}^{[\mathrm{FB1}]}\big) = \mathbb{P}\big\{y_{i_1j_1}^{[\mathrm{FB1}]}y_{i_2j_2}^{[\mathrm{FB1}]} = 1\big\}$ via

$$\begin{aligned}
&\frac{1}{n!}\binom{n}{j_2-i_1+2}\binom{j_2-i_1+1}{j_2-i_2+2}(j_2-i_2+1)!(i_2-i_1-1)!(n-(j_2-i_1+2))!\\
&= \frac{1}{(j_2-i_1+2)(j_2-i_2+2)}.
\end{aligned}$$

All permutations with $y_{i_1j_1}^{[\mathrm{FB1}]}y_{i_2j_2}^{[\mathrm{FB1}]} = 1$ can be obtained in this case in the following way: Choose from the given set $\{1,\dots,n\}$ exactly j_2-i_1+2 elements, which will be put into positions $i_1,\dots,j_2$ resp. position 1 in the way described later on. Of course, the remaining $n-(j_2-i_1+2)$ elements can be put into the remaining positions in $(n-(j_2-i_1+2))!$ different ways. From the chosen elements we select the smallest and put it into position i_1. Next we choose from the j_2-i_1+1 remaining elements again a set of j_2-i_2+2 elements. The smallest element of this set is put into position i_2 and the other j_2-i_2+1 elements are put in arbitrary order ($(j_2-i_2+1)!$ possibilities) into positions 1 resp. $i_2+1,\dots,j_2$. The i_2-i_1-1 elements that are not chosen in the second round are put in one of $(i_2-i_1-1)!$ different ways into the positions $i_1+1,\dots,i_2-1$. This leads to the probability as given above. Similar considerations for the remaining cases give the above formula.

We also show for this example how to simplify the sums that are appearing:

$$\begin{aligned}
&\sum_{2\le i_1<i_2\le j_1<j_2\le n}\frac{1}{(j_2-i_1+2)(j_2-i_2+2)}\\
&=\sum_{i_1=2}^{n-2}\sum_{i_2=i_1+1}^{n-1}\sum_{j_2=i_2+1}^{n}\sum_{j_1=i_2}^{j_2-1}\frac{1}{(j_2-i_1+2)(j_2-i_2+2)}\\
&=\sum_{i_1=2}^{n-2}\sum_{j_2=i_1+2}^{n}\sum_{i_2=i_1+1}^{j_2-1}\frac{1}{j_2-i_1+2}\left(1-\frac{2}{j_2-i_2+2}\right)\\
&=\sum_{i_1=2}^{n-2}\sum_{j_2=i_1+2}^{n}\frac{1}{j_2-i_1+2}\left(j_2-i_1-1-2\Big(H_{j_2-i_1+1}-\frac{3}{2}\Big)\right)\\
&=\sum_{i_1=2}^{n-2}\sum_{j_2=i_1+2}^{n}\left(1-2\frac{H_{j_2-i_1+2}}{j_2-i_1+2}+\frac{2}{(j_2-i_1+2)^2}\right)\\
&=\sum_{i_1=2}^{n-2}\Big(n-i_1-1-H_{n-i_1+2}^2-H_{n-i_1+2}^{(2)}+H_3^2+H_3^{(2)}+2H_{n-i_1+2}^{(2)}-2H_3^{(2)}\Big)\\
&=\sum_{k=1}^{n-3}k-\sum_{k=4}^{n}H_k^2+\sum_{k=4}^{n}H_k^{(2)}+(H_3^2-H_3^{(2)})(n-3)\\
&=\frac{(n-3)(n-2)}{2}-(n+1)H_n^2+(2n+1)H_n-2n+4H_3^2-7H_3+6\\
&\quad+(n+1)H_n^{(2)}-H_n-4H_3^{(2)}+H_3+(n-3)(H_3^2-H_3^{(2)});
\end{aligned}$$

we used here and for the remaining cases several identities involving harmonic numbers to simplify the sums that occur. The identities in question are

$$\sum_{k=1}^{n} H_k = (n+1)H_n - n, \qquad \sum_{k=1}^{n} H_k^{(2)} = (n+1)H_n^{(2)} - H_n,$$

$$\sum_{k=1}^{n} \frac{H_k}{k} = \frac{1}{2}(H_n^2 + H_n^{(2)}), \qquad \sum_{k=1}^{n} kH_k = \frac{n(n+1)}{2}H_n - \frac{n(n-1)}{2},$$

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

These identities are all known. Once such an identity is known, it can easily be proved by induction. Of course, generating functions also help. As a simple example consider

$$\sum_{n\geq 1} H_n z^n = \frac{1}{1-z}\log\frac{1}{1-z};$$

from this explicit form various others can be derived by standard manipulations, such as differentiations, integrations, etc. Of course, they can all be derived with C. Schneider's software Σ, see [7].

By simplifying the remaining sums in the same manner, we get a closed form for the second moment. Subtracting the square of the expectation from that we finally find that the variance $\mathbb{V}(a^{[\mathrm{FB1}]}) = \mathbb{E}\big((a^{[\mathrm{FB1}]})^2\big) - \big(\mathbb{E}(a^{[\mathrm{FB1}]})\big)^2$ is given by

$$\mathbb{V}(a^{[\mathrm{FB1}]}) = -(2n+3)H_n^2 - (n+1)^2H_n^{(2)} + (3n-2)H_n + \frac{31}{12}n^2 + \frac{11}{2}n - \frac{1}{12}, \qquad (10)$$

and asymptotically by

$$\mathbb{V}(a^{[\mathrm{FB1}]}) = \Big(\frac{31}{12} - \frac{\pi^2}{6}\Big)n^2 + \mathcal{O}(n\log^2 n). \qquad (11)$$

Analogous computations give the variance $\mathbb{V}(\kappa^{[\mathrm{FB1}]})$:

$$\mathbb{V}(\kappa^{[\mathrm{FB1}]}) = -(2n+3)H_n^2 - (n+1)^2H_n^{(2)} + (n-4)H_n + 3n(n+3), \qquad (12)$$

and asymptotically

$$\mathbb{V}(\kappa^{[\mathrm{FB1}]}) = \Big(3 - \frac{\pi^2}{6}\Big)n^2 + \mathcal{O}(n\log^2 n). \qquad (13)$$

3. Analysis of Keller's heuristic 2

Heuristic 2 for cycle leader checking (called `Fastbreak 2`) is described by Keller as follows:

If only r elements have not been visited yet by function ***PermuteCycle****, then proceeding along a cycle is only useful for r steps. Otherwise, the cycle currently traversed must contain an element that has already been visited by function* ***PermuteCycle****, the cycle hence has already been permuted, and i cannot be a leader.*

An implementation of this heuristic 2 was given by Keller (see [3, Fig. 4]), and is here also given as Algorithm 2.

It should be remarked that this *implementation* of heuristic 2 does *not* make full usage of the power of this rule, since in the implementation, the cycle leader check is not stopped (if not earlier by other conditions) until the number of steps that are done

Algorithm 2 Keller's cycle leader checking algorithm from [3] for heuristic 2.

Input: $\pi = (\pi(1), \ldots, \pi(n))$ permutation of length n,
$1 \leq i \leq n$, $0 \leq \text{remaining} \leq n$ integers.
Output: leader $\in \{\text{true}, \text{false}\}$, leader = true iff i is a cycle leader.

```
j ← i
leader ← true
steps ← 0
repeat
  j ← π(j)
  steps ← steps + 1
  if (j < i) or (steps > remaining) then
    leader ← false
    break
  end if
until j = i
return leader
```

checking a certain element is larger than the number of remaining steps. However, one could already leave when the number of steps equals the number of remaining steps. Then we either found a cycle of the length of the number of remaining steps, and we call the permuting algorithm, or we know then that the checked element is not a cycle leader. A *modified* cycle leader checking algorithm, which uses this principle is given as Algorithm 3. It is assumed, that each implementation (Algorithm 2 or Algorithm 3) is used together with the permutation algorithm given in [3, Fig. 3]. Then it is guaranteed, that the case, where (in above described heuristic 2) $r = 0$ elements have not been visited, is also covered (this is exactly heuristic 1). We implemented all algorithms in `C` as well since all algorithms given by Keller are implemented in `C`. These versions can be obtained from the authors. We replaced Keller's "`TestHypo(j,i)`" by "$j \geq i$" since that is exactly what it is. Our average-case analysis is based on this improved implementation.

In Keller's implementation of heuristic 2 as given by Algorithm 2, more function calls are made on average, and although this has asymptotically no influence to the main term, the savings are quite noteworthy for smaller sizes. The smallest permutation where a difference between both implementations occurs is (4)(123). Detecting element 1 as a cycle leader requires in both implementations 3 function calls, but by checking the element 2 for a possible cycle leader, one could stop after one step (it is already known, that only a cycle of length one is missing), but Keller's implementation uses 2 steps. Since by checking the elements 3 and 4 for possible cycle leaders, both implementations make one function call for each element, we obtain that Algorithm 3 requires here 6 function calls whereas Algorithm 2 requires 7 function call. Since all the experimental results in [3] are given for his implementation, we briefly analyze this version also.

Now we come to the average-case analysis of the algorithm with heuristic 2 as implemented by Algorithm 3. We first study the parameter κ (the number of function calls) in detail, showing how to compute $\mathbb{E}(\kappa^{[\mathrm{FB2}]})$.

For Knuth's parameter a (the number of passes of the `while`-loop), we only give the formula for the expectation and omit the analogous computations to produce $\mathbb{E}(a^{[\mathrm{FB2}]})$.

Algorithm 3 Keller's cycle leader checking algorithm from [3] for heuristic 2 with improvement as discussed in the text.

Input: $\pi = (\pi(1), \ldots, \pi(n))$ permutation of length n,
$1 \le i \le n$, $0 \le \text{remaining} \le n$ integers.
Output: leader $\in \{\text{true}, \text{false}\}$, leader $=$ true iff i is a cycle leader.

```
leader ← true
j ← π(i)
steps ← 1
while (j > i) and (steps < remaining) do
    j ← π(j)
    steps ← steps + 1
end while
if j ≠ i then
    leader ← false
end if
return leader
```

The third variation that we briefly discuss at the end of this section, as the computations are somewhat similar, is the study of the number of function calls in Keller's implementation given by Algorithm 2. We will also obtain a formula for $\mathbb{E}(\tilde{\kappa}^{[\mathrm{FB2}]})$.

If $q_1 \ldots q_n$ is the (canonical) cycle representation of a given permutation π of length n, then we get the following number $\kappa^{[\mathrm{FB2}]}$ of function calls:

$$\kappa^{[\mathrm{FB2}]} = \#\Big\{(i,j) \;\big|\; 1 \le i \le j \le n,\ (q_i \le q_{i+1}, q_i \le q_{i+2}, \ldots, q_i \le q_j) \textbf{ and } \Big[\Big(j - i + 2 \le \min_{k<i} \{k \mid q_k < q_i \textbf{ and } q_1 > q_i, \ldots, q_{k-1} > q_i\}\Big) \textbf{ or } \Big(q_1 > q_i, \ldots, q_{i-1} > q_i\Big)\Big]\Big\}. \quad (14)$$

Then a first formula for the expected value $\mathbb{E}(\kappa^{[\mathrm{FB2}]})$ of this parameter is given by the following expression:

$$\mathbb{E}(\kappa^{[\mathrm{FB2}]}) = \sum_{1 \le i \le j \le n} \frac{1}{n!} \sum_{L=1}^{n} \Bigg[\binom{n-L}{j-1}(j-1)!(n-j)! + \sum_{k=j-i+2}^{i-1} \binom{n-L}{j-i+k-1}(j-i+k-1)!(L-1)(n-k-(j-i)-1)!\Bigg]. \quad (15)$$

It is obtained by summing up the expectations of the indicator variable which is true iff the function call $\pi(q_j)$ is made when testing q_i as a possible cycle leader for $1 \le i \le j \le n$. This means of course that at least $j - i + 1$ function calls are done to check whether q_i is a cycle leader. This happens only if $q_{i+1} > q_i, \ldots, q_j > q_i$ and either q_i is actually a cycle leader ($q_1 > q_i, \ldots, q_{i-1} > q_i$) or the number of steps $j - i + 1$ is not larger than the number of remaining steps, which is given by $k - 1$, where k is the position of the largest cycle leader q_k being smaller than q_i ($q_k < q_i$ **and** $q_1 > q_i, \ldots, q_{k-1} > q_i$). If one distinguishes the possible values of $L := q_i$ the formula (15) follows.

In the following we want to get a closed formula for the expression (15). We consider first the simpler part of this expression and get

$$\sum_{i=1}^{n}\sum_{j=i}^{n}\frac{1}{n!}\sum_{L=1}^{n}\binom{n-L}{j-1}(j-1)!(n-j)! = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=i}^{n}\frac{1}{\binom{n-1}{j-1}}\sum_{L=1}^{n}\binom{n-L}{j-1}$$
$$= \frac{1}{n}\sum_{i=1}^{n}\sum_{j=i}^{n}\frac{\binom{n}{j}}{\binom{n-1}{j-1}} = \sum_{i=1}^{n}\sum_{j=i}^{n}\frac{1}{j} = \sum_{j=1}^{n}\frac{1}{j}\sum_{i=1}^{j}1 = n.$$

For the other part of expression (15) we get first

$$\sum_{i=1}^{n}\sum_{j=i}^{n}\frac{1}{n!}\sum_{L=1}^{n}\sum_{k=j-i+2}^{i-1}\binom{n-L}{j-i+k-1}(j-i+k-1)!(L-1)(n-k-(j-i)-1)!$$
$$= \frac{1}{n(n-1)}\sum_{i=1}^{n}\sum_{j=i}^{n}\sum_{k=j-i+2}^{i-1}\frac{1}{\binom{n-2}{j-i+k-1}}\sum_{L=1}^{n}\binom{n-L}{j-i+k-1}(L-1)$$
$$= \frac{1}{n-1}\sum_{i=1}^{n}\sum_{j=i}^{n}\sum_{k=j-i+2}^{i-1}\frac{1}{\binom{n-2}{j-i+k-1}}\sum_{L=1}^{n}\binom{n-L}{j-i+k-1}$$
$$-\frac{1}{n}\sum_{i=1}^{n}\sum_{j=i}^{n}\sum_{k=j-i+2}^{i-1}\frac{1}{\binom{n-1}{j-i+k}}\sum_{L=1}^{n}\binom{n-L+1}{j-i+k}$$
$$= \frac{1}{n-1}\sum_{i=1}^{n}\sum_{j=i}^{n}\sum_{k=j-i+2}^{i-1}\frac{\binom{n}{j-i+k}}{\binom{n-2}{j-i+k-1}} - \frac{1}{n}\sum_{i=1}^{n}\sum_{j=i}^{n}\sum_{k=j-i+2}^{i-1}\frac{\binom{n+1}{j-i+k+1}}{\binom{n-1}{j-i+k}}$$
$$= \sum_{i=1}^{n}\sum_{j=i}^{n}\sum_{k=j-i+2}^{i-1}\left(\frac{n}{(j-i+k)(n-(j-i)-k)} - \frac{n+1}{(j-i+k+1)(n-(j-i)-k)}\right)$$
$$= \sum_{i=1}^{n}\sum_{l=0}^{n-i}\sum_{k=l+2}^{i-1}\left(\frac{1}{l+k} - \frac{1}{l+k+1}\right) = \sum_{i=3}^{n}\sum_{l=0}^{\min(i-3,n-i)}\sum_{k=l+2}^{i-1}\left(\frac{1}{l+k} - \frac{1}{l+k+1}\right)$$
$$= \sum_{i=3}^{n}\sum_{l=0}^{\min(i-3,n-i)}\left(\frac{1}{2l+2} - \frac{1}{l+i}\right).$$

This sum can be further simplified to

$$\sum_{i=3}^{n}\sum_{l=0}^{\min(i-3,n-i)}\left(\frac{1}{2l+2} - \frac{1}{l+i}\right) = \sum_{l=0}^{\lfloor\frac{n+1}{2}\rfloor-2}\sum_{i=l+3}^{n-l}\left(\frac{1}{2l+2} - \frac{1}{l+i}\right)$$
$$= \sum_{l=0}^{\lfloor\frac{n+1}{2}\rfloor-2}\left(\frac{n}{2l+2} - 1 - H_n + H_{2l+2}\right)$$
$$= \left(\left\lfloor\frac{n-1}{2}\right\rfloor + \frac{1}{2}\right)H_{2\lfloor\frac{n-1}{2}\rfloor} + \left(\frac{n}{2} + \frac{1}{4}\right)H_{\lfloor\frac{n-1}{2}\rfloor} - \left\lfloor\frac{n-1}{2}\right\rfloor H_n - 2\left\lfloor\frac{n-1}{2}\right\rfloor,$$

where we used

$$\sum_{k=1}^{n} H_{2k} = \Big(n+\frac{1}{2}\Big)H_{2n} + \frac{1}{4}H_n - n.$$

Adding both parts, we get for the expected number $\mathbb{E}(\kappa^{[\mathrm{FB2}]})$ of function calls

$$\mathbb{E}(\kappa^{[\mathrm{FB2}]}) = \Big(\Big\lfloor\frac{n-1}{2}\Big\rfloor + \frac{1}{2}\Big)H_{2\lfloor\frac{n-1}{2}\rfloor} + \Big(\frac{n}{2}+\frac{1}{4}\Big)H_{\lfloor\frac{n-1}{2}\rfloor} - \Big\lfloor\frac{n-1}{2}\Big\rfloor H_n - 2\Big\lfloor\frac{n-1}{2}\Big\rfloor + n.$$

It can also be written as

$$\mathbb{E}(\kappa^{[\mathrm{FB2}]}) = \Big(\Big\lfloor\frac{n}{2}\Big\rfloor + \frac{1}{4}\Big)H_{\lfloor\frac{n}{2}\rfloor} + \frac{1}{2}H_n + \frac{1}{2}\left(\Big\lceil\frac{n}{2}\Big\rceil - \Big\lfloor\frac{n}{2}\Big\rfloor\right)\left(H_{\lfloor\frac{n}{2}\rfloor}+1\right), \tag{16}$$

and is asymptotically given by

$$\mathbb{E}(\kappa^{[\mathrm{FB2}]}) = \frac{1}{2}n\log n + \mathcal{O}(n). \tag{17}$$

For Knuth's parameter $a^{[\mathrm{FB2}]}$, defined by

$$a^{[\mathrm{FB2}]} = \#\Big\{(i,j) \,\Big|\, 1 \le i < j \le n,\ (q_i < q_{i+1}, q_i < q_{i+2}, \dots, q_i < q_j)\ \textbf{and}\ \Big[\Big(j-i+2 \le \min_{k<i}\{k \mid q_k < q_i\ \textbf{and}\ q_1 > q_i, \dots, q_{k-1} > q_i\}\Big)\ \textbf{or}\ \Big(q_1 > q_i, \dots, q_{i-1} > q_i\Big)\Big]\Big\}, \tag{18}$$

we get in the same style the expectation

$$\mathbb{E}(a^{[\mathrm{FB2}]}) = \frac{2n+1}{4}H_{\lfloor\frac{n+1}{2}\rfloor} + \frac{1}{2}H_{2\lfloor\frac{n+1}{2}\rfloor} - \frac{1}{2}\Big(\Big\lfloor\frac{n+1}{2}\Big\rfloor - \Big\lfloor\frac{n}{2}\Big\rfloor\Big) - \frac{n+1}{2}. \tag{19}$$

In Keller's implementation of heuristic 2, the number $\tilde{\kappa}^{[\mathrm{FB2}]}$ of function calls for a given permutation π of length n can be described by

$$\tilde{\kappa}^{[\mathrm{FB2}]} = \#\Big\{(i,j) \,\Big|\, 1 \le i < j \le n,\ (q_i < q_{i+1}, q_i < q_{i+2}, \dots, q_i < q_j)\ \textbf{and}\ \Big[\Big(j-i+1 \le \min_{k<i}\{k \mid q_k < q_i\ \textbf{and}\ q_1 > q_i, \dots, q_{k-1} > q_i\}\Big)\ \textbf{or}\ (q_1 > q_i, \dots, q_{i-1} > q_i)\Big]\Big\} + \#\{i \mid q_i \le q_1\}. \tag{20}$$

The expected value $\mathbb{E}(\tilde{\kappa}^{[\mathrm{FB2}]})$ is then given by the following expression:

$$\mathbb{E}(\tilde{\kappa}^{[\mathrm{FB2}]}) = \sum_{1\le i<j\le n}\frac{1}{n!}\sum_{L=1}^{n}\Bigg[\binom{n-L}{j-1}(j-1)!(n-j)! + \sum_{k=j-i+1}^{i-1}\binom{n-L}{j-i+k-1}(j-i+k-1)!(L-1)(n-k-(j-i)-1)!\Bigg] + \frac{1}{2}(n-1)+1. \tag{21}$$

One computes

$$\sum_{i=1}^{n}\sum_{j=i+1}^{n}\frac{1}{n!}\sum_{L=1}^{n}\binom{n-L}{j-1}(j-1)!(n-j)! = n - H_n,$$

$$\sum_{i=1}^{n}\sum_{j=i+1}^{n}\frac{1}{n!}\sum_{L=1}^{n}\sum_{k=j-i+1}^{i-1}\binom{n-L}{j-i+k-1}(j-i+k-1)!(L-1)(n-k-(j-i)-1)!$$

$$= \sum_{i=2}^{n}\sum_{l=1}^{\min(i-2,n-i)}\left(\frac{1}{2l+1}-\frac{1}{l+i}\right) = \sum_{l=1}^{\lfloor\frac{n}{2}\rfloor-1}\sum_{i=l+2}^{n-l}\left(\frac{1}{2l+1}-\frac{1}{l+i}\right)$$

$$\overset{n\geq 1}{=} \left(n+\left\lfloor\frac{n}{2}\right\rfloor+\frac{1}{2}\right)H_{2\lfloor\frac{n}{2}\rfloor-1} - \left(\frac{n}{2}+\frac{1}{4}\right)H_{\lfloor\frac{n}{2}\rfloor-1} - \left(\left\lfloor\frac{n}{2}\right\rfloor-1\right)H_n - 2\left\lfloor\frac{n}{2}\right\rfloor - n + \frac{1}{2},$$

where we used for the last equation the formula

$$\sum_{k=0}^{n} H_{2k+1} = \left(n+\frac{3}{2}\right)H_{2n+1} - n - \frac{1}{2} - \frac{1}{4}H_n.$$

Adding all parts, one gets a formula for $\mathbb{E}(\tilde{\kappa}^{[\mathrm{FB2}]})$, which can be simplified to

$$\mathbb{E}(\tilde{\kappa}^{[\mathrm{FB2}]}) = \frac{1}{2}(n+1)H_n - \frac{n}{2} + \frac{1}{2} + \frac{n}{2}H_{2\lfloor\frac{n}{2}\rfloor} - \left(\frac{n}{2}+\frac{1}{4}\right)H_{\lfloor\frac{n}{2}\rfloor}, \tag{22}$$

and is asymptotically again given by

$$\mathbb{E}(\tilde{\kappa}^{[\mathrm{FB2}]}) = \frac{1}{2}n\log n + \mathcal{O}(n). \tag{23}$$

4. Experimental results compared to the mathematical formulæ

In order to compare our theoretical findings of the expected savings due to heuristics 1 resp. 2, we also produced some experimental results. We used the computer algebra system Maple and generated for each $n = 2^5, 2^6, \ldots, 2^{20}$ with the `randperm` function of the `combinat` package 50 random permutations of length n and counted the number of function calls within the cycle leader checking algorithm for each of the analyzed variants.

Figures 1, 2 and 3 show the savings of function calls using heuristic 1, heuristic 2 (the improved implementation given in this paper) and heuristic 2 (implementation by Keller), compared to the standard algorithm. In the figures the horizontal bars mark for each n the mean value of the experimental result, whereas the boxes mark the theoretical average savings, which are obtained from our formulæ κ, $\kappa^{[\mathrm{FB1}]}$, $\kappa^{[\mathrm{FB2}]}$ resp. $\tilde{\kappa}^{[\mathrm{FB2}]}$.

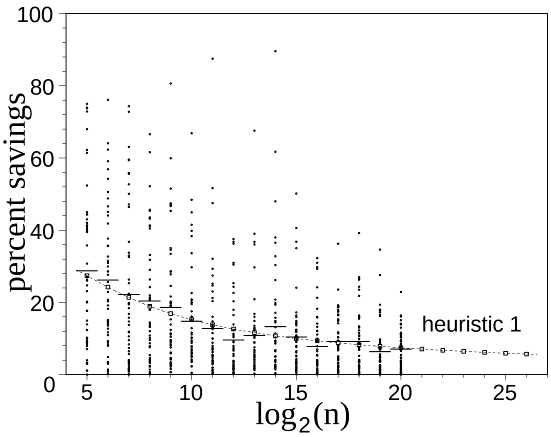

FIGURE 1. Expected savings of function calls with heuristic 1.

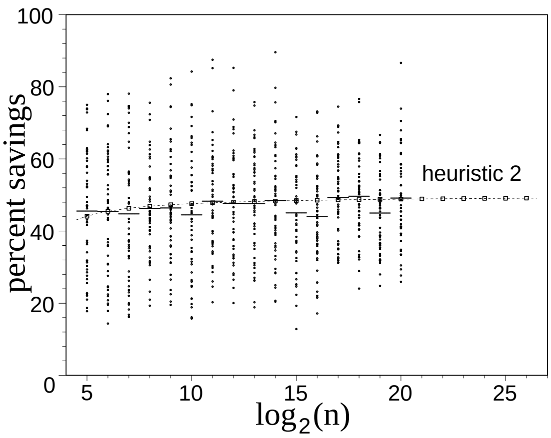

FIGURE 2. Expected savings of function calls with heuristic 2 (implementation presented in this paper).

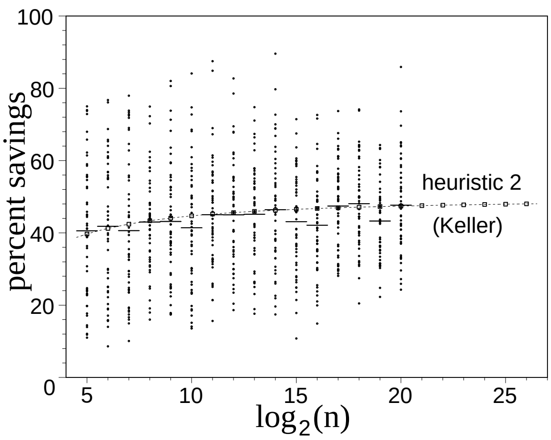

FIGURE 3. Expected savings of function calls with heuristic 2 (Keller's implementation).

Our theoretical results fit quite well with the experimental results presented here and also with the experimental results given by Keller, although only relatively small test sets are used (compared to the huge number of permutations of length 2^{20}, say).

For the reader's convenience, we collect our major findings in a table.

quantity	name	def.	exact formula	asympt.
Knuth's parameter "a"	a	(1)	$(n+1)H_n - 2n$	$n \log n$
Keller's version "κ"	κ	(3)	$(n+1)H_n - n$	$n \log n$
"a" with `Fastbreak 1`	$a^{[\mathrm{FB1}]}$	(6)	$(n+2)H_n - \frac{5n}{2} - \frac{1}{2}$	$n \log n$
"κ" with `Fastbreak 1`	$\kappa^{[\mathrm{FB1}]}$	(8)	$(n+2)H_n - 2n$	$n \log n$
"a" with `Fastbreak 2`	$a^{[\mathrm{FB2}]}$	(18)	$\frac{2n+1}{4} H_{\lfloor \frac{n+1}{2} \rfloor} + \frac{1}{2} H_{2\lfloor \frac{n+1}{2} \rfloor}$ $-\frac{1}{2}(\lfloor \frac{n+1}{2} \rfloor - \lfloor \frac{n}{2} \rfloor) - \frac{n+1}{2}$	$\frac{1}{2} n \log n$
"κ" with `Fastbreak 2`	$\kappa^{[\mathrm{FB2}]}$	(14)	$(\lfloor \frac{n}{2} \rfloor + \frac{1}{4}) H_{\lfloor \frac{n}{2} \rfloor} + \frac{1}{2} H_n$ $+\frac{1}{2}(\lceil \frac{n}{2} \rceil - \lfloor \frac{n}{2} \rfloor)(H_{\lfloor \frac{n}{2} \rfloor} + 1)$	$\frac{1}{2} n \log n$
Keller's implementation of "κ" with `Fastbreak 2`	$\tilde{\kappa}^{[\mathrm{FB2}]}$	(20)	$(n + \lfloor \frac{n}{2} \rfloor + \frac{1}{2}) H_{2\lfloor \frac{n}{2} \rfloor - 1}$ $-(\frac{n}{2} + \frac{1}{4}) H_{\lfloor \frac{n}{2} \rfloor - 1} - (\lfloor \frac{n}{2} \rfloor - 1) H_n$ $-2\lfloor \frac{n}{2} \rfloor - n + \frac{1}{2}$	$\frac{1}{2} n \log n$

FIGURE 4. The averages of the parameters studied in this paper.

Acknowledgment. The comments of two referees helped to improve the presentation.

REFERENCES

1. F. E. Fich, J. I. Munro, and P. V. Poblete, *Permuting in place*, SIAM Journal on Computing **24** (1995), 266–278.
2. H.-K. Hwang and R. Neininger, *Phase change of limit laws in the quicksort recurrence under varying toll functions*, SIAM Journal on Computing **31** (2002), 1687–1722.
3. J. Keller, *A heuristic to accelerate in–situ permutation algorithms*, Information Processing Letters **81** (2002), 119–125.
4. P. Kirschenhofer, H. Prodinger, and R. F. Tichy, *A contribution to the analysis of in situ permutation*, Glasnik Mathematicki **22(42)** (1987), 269–278.
5. D. E. Knuth, *Mathematical analysis of algorithms*, Information Processing 71, North Holland Publishing Company, 1972, Proceedings of IFIP Congress, Ljubljana, 1971, pp. 19–27. Reprinted in: *Selected papers on analysis of algorithms*, CSLI Publications, Stanford, CA, 2000.
6. H. Prodinger, *Combinatorics of geometrically distributed random variables: Inversions and a parameter of Knuth*, Annals of Combinatorics **5** (2001), 241–250.

7. C. Schneider, *Symbolic summation in difference fields*, PhD thesis, Univ. Linz, 2001.
8. R. Sedgewick and P. Flajolet, *An introduction to the analysis of algorithms*, Addison-Wesley, 1996.

Alois Panholzer, Institut für Algebra und Computermathematik, TU Wien, Wiedner Hauptstr. 8-10, 1040 Wien, Austria
E-mail address: Alois.Panholzer@tuwien.ac.at

Helmut Prodinger, The John Knopfmacher Centre for Applicable Analysis and Number Theory, School of Mathematics, University of the Witwatersrand, P. O. Wits, 2050 Johannesburg, South Africa
E-mail address: helmut@staff.ms.wits.ac.za

Marko Riedel, EDV, Neue Arbeit gGmbH, Presselstr. 29, 70191 Stuttgart, Germany
E-mail address: mriedel@neuearbeit.de