<stupid acronym>- An algorithm for faster sock sorting

CelloClemens[1,2], Henriente[1],

November 22, 2022

[1]Department for theoretical laundry science, Karlsruhe institute of suffering and sorrow (KISS), Karlsruhe, Germany
[2]Institute of laundry sorting, Department for socks, Karlsruhe institute of suffering and sorrow (KISS), Karlsruhe, Germany

Abstract Sorting socks can often be a time consuming task. This paper introduces the fastest method known in the scientific community to tackle this challanging task. To be able to implement this new algorithm a new data structure will be introduced and discussed. Abundant application of this novel algorithm may be able to reduce the time required for sorting socks considerably.

1 Introduction

While sorting algorithms are one of the most discussed algorithms in the computer science community, application of this field to laundry is still quite new. In fact no research is known to the authors connecting the fields of computer science and laundry sorting. A few definitions are required in order to establish a baseline for the algorithm discussed in the following paper.

1.1 Definitions

In this section a few definitions, common in the field of theoretical laundry science shall be introduced. These are required to understand the algorithm and its advantages.

1.1.1 Sock

Let Λ_a be the Set of laundry. The set of socks, $\Sigma \subset \Lambda_a$ is defined as $\Sigma := \{s \in \Lambda_a | \chi(s) = 1\}$[1], where $\chi(s)$ is the Euler characteristic of s. For every sock s there is an equal counterpart s^{-1} giving rise to the identity $s \cong s^{-1}$. The task commonly known as "sock sorting" is in fact the search for this isomorphism η and matching every sock s to its inverse s^{-1}.

1.1.2 Laundry basket

Let $\Lambda \subseteq \Lambda_a$ be a set of laundry items. Then a laundry basket is a triple $(\Lambda, +, -)$ representing a data structure that implements the following functions:

- `get`: $\mathscr{L} \in \Lambda_a$, returns a uniformly random laundry item from the basket or $\mathscr{L}_0$, the Zero element of laundry, iff There are no items left.

[1]Yes, some socks have holes. So what?!

Fig. 1: Three single socks.

- `put` $(\mathscr{L} \in \Lambda_a)$, deposits the given laundry item into the basket.

Note that both operations run in $\mathcal{O}(1)$. Because of the nature of a laundry basket finding a unique item requires transferring the content of the whole basket to a new basket thus requiring $\mathcal{O}(n)$ operations, n being the number of items currently inside the basket.

1.2 Ongoing and latest research

To fully appreciate the gravity of <stupid acronym>it has to first be discussed how most resent research tackles the problem of sock sorting. The following code describes the most recently developed sock sorting algorithm from the paper by **My colleague et al.** Which is the current industry standard. Notice whe code has a runtime complex-

ity of $\mathcal{O}(n^2)$.

Algorithm 1 Conventional sock sorting

```
▷ initialize a new laundry basket with a given set of laundry
A ← Λ   ▷ WLOG assume ∀𝓛 ∈ Λ|𝓛 is a sock
matches ← []
repeat
    𝓛 ← A.get
    repeat   ▷ find the inverse Sock by checking all other socks
        𝓛′ ← A.get
    until 𝓛′ = 𝓛^−1
    matches.append((𝓛, 𝓛′))
until 𝓛 ≠ 𝓛_0
```

2 Methodology

3 Discussion and Results

4 Conclusion

5 Acknowledgements

I did this all by myself, so I'm kinda awesome. But I guess I hocked and edited this template from the cowshed article so thanks for that William Roper