Abstract

In this paper, we study testing of sequence properties that are defined by forbidden order patterns. A sequence $f : \{1, \ldots, n\} \to \mathbb{R}$ of length n contains a pattern $\pi \in \mathfrak{S}_k$ ($\mathfrak{S}_k$ is the group of permutations of k elements), iff there are indices $i_1 < i_2 < \cdots < i_k$, such that $f(i_x) > f(i_y)$ whenever $\pi(x) > \pi(y)$. If f does not contain π, we say f is π-free. For example, for $\pi = (2, 1)$, the property of being π-free is equivalent to being non-decreasing, i.e. monotone. The property of being $(k, k-1, \ldots, 1)$-free is equivalent to the property of having a partition into at most $k-1$ non-decreasing subsequences.

Let $\pi \in \mathfrak{S}_k$, k constant, be a (forbidden) pattern. Assuming f is stored in an array, we consider the property testing problem of distinguishing the case that f is π-free from the case that f differs in more than ϵn places from any π-free sequence. We show the following results: There is a clear dichotomy between the monotone patterns and the non-monotone ones:

- For monotone patterns of length k, i.e., $(k, k-1, \ldots, 1)$ and $(1, 2, \ldots, k)$, we design *non-adaptive* one-sided error ϵ-tests of $(\epsilon^{-1} \log n)^{O(k^2)}$ query complexity.
- For non-monotone patterns, we show that for any size-k non-monotone π, *any non-adaptive* one-sided error ϵ-test requires at least $\Omega(\sqrt{n})$ queries. This general lower bound can be further strengthened for specific non-monotone k-length patterns to $\Omega(n^{1-2/(k+1)})$. On the other hand, there always exists a *non-adaptive* one-sided error ϵ-test for $\pi \in \mathfrak{S}_k$ with $O(\epsilon^{-1/k} n^{1-1/k})$ query complexity. Again, this general upper bound can be further strengthened for specific non-monotone patterns. E.g., for $\pi = (1, 3, 2)$, we describe an ϵ-test with (almost tight) query complexity of $\widetilde{O}(\sqrt{n})$.

Finally, we show that adaptivity can make a big difference in testing non-monotone patterns, and develop an *adaptive* algorithm that for any $\pi \in \mathfrak{S}_3$, tests π-freeness by making $(\epsilon^{-1} \log n)^{O(1)}$ queries. For all algorithms presented here, the running times are linear in their query complexity.