

## РЕАЛІЗАЦІЯ ТА ПРАКТИЧНЕ ЗАСТОСУВАННЯ ВИЧЕРПНИХ МАСИВІВ В РОЗРОБЦІ ПРИКЛАДНИХ ІНТЕРФЕЙСІВ ПРОГРАМУВАННЯ

Авідзба В.Ю.

e-mail: vladyslav.avidzba@nure.ua

Харківський національний університет радіоелектроніки, каф. СТ  
м. Харків, Україна

This work is devoted to studying the exhaustive array problem in Typescript. The exhaustive array problem has been already discovered and one of its solutions is known. However the known solution is impractical due to its factorial complexity. This work proves that more than one exhaustive array exists, classifies them by 5 orders and provides a description of the algorithms for exhaustive arrays of the II and III order. This work also compares the obtained algorithms with the existing one, showing that it may have practical use cases as its algorithmic complexity is polynomial rather than factorial.

Проблема вичерпних масивів в мові програмування Typescript полягає у відсутності у мові програмування вбудованих інструментів співставлення елементів певного об'єднання значень (union-у) масиву елементів, що за певними правилами складається з конститuent цього об'єднання.

На поточний момент ця проблема відома і має наступне формулювання: «необхідно створити такий масив ExhaustiveArray<U>, який складається з усіх конститuent U, при чому кожна конститuenta входить масив лише один раз». Рішення цієї задачі в такому формулюванні відоме, але це рішення має лише теоретичну цінність та на практиці не використовується через факторіальну складність алгоритму.

Визначається наступна класифікація вичерпних масивів:

1) Вичерпний масив 0 порядку – масив довільних елементів довільної довжини. Форма запису мовою Typescript: any[].

2) Вичерпний масив I порядку – масив елементів типу T довільної довжини. Форма запису мовою Typescript: T[].

3) Вичерпний масив II порядку – масив елементів типу T такої довжини, яка дорівнює кількості конститuent («довжині») union-у U. Цей масив не можна виразити в «елементарних» лексемах мови Typescript.

4) Вичерпний масив III порядку – масив елементів типу U такої самої довжини, яка дорівнює кількості конститuent («довжині») union-у U. Цей масив не можна виразити в «елементарних» лексемах мови Typescript.

5) Вичерпний масив IV порядку – масив елементів типу U такої самої довжини, яка дорівнює, яка дорівнює кількості конститuent («довжині») union-у U, та кожна конститuenta має входити рівно один раз. Цей масив не можна виразити в «елементарних» лексемах мови Typescript.

Стратегія пошуку алгоритму вичерпного масиву III порядку полягає на представленні масиву як запису  $\text{Record}\langle 0 \mid 1 \mid 2 \mid \dots \mid \text{Length}\langle U \rangle - 1, U \rangle$  &  $\text{Array}\langle U \rangle$ . Вся складність алгоритму полягає в обчисленні  $\text{Length}\langle U \rangle$ .

Рішення задачі про вичерпний масив III порядок ґрунтується на ідеї, що в Typescript, як в над множині Javascript наявна структурна типізація, та що така структура даних як масив в Javascript структурно еквівалентна об'єкту, ключі якого є числами, а також в якого присутня властивість `length`. А для об'єктів в Typescript засіб вичерпного співставлення конститuent `union-у` з певними значеннями через вбудований тип `Record`. Таким чином задача зводиться до перетворення вихідного типу `union-у U` в  $\text{Record}\langle 0 \mid 1 \mid 2 \mid \dots \mid N - 1, U \rangle$ , де  $N$  – це «довжина» `union-у U`.

Задача зводиться до проблеми про перетворення `union-у` в кортеж, бо властивістю довжини в мові TS володіють лише кортежі. Рішення задачі про перетворення об'єднання в кортеж є загальновідомим. Отримавши алгоритм вичерпного масиву III порядку, дуже легко показати, що також отримано алгоритм вичерпного масиву II порядку, бо масив III порядку є окремим випадком масиву II порядку.

Для порівняння отриманий алгоритм з існуючим алгоритмом IV порядку і обґрунтування його практичної цінності в використанні треба порівнювати їх за асимптотичною складністю алгоритму. Отриманий алгоритм має обмеження в використанні через особливості роботи рекурсії мови Typescript, при цьому на відміну від відомого алгоритму IV порядку отриманий алгоритм є практично придатним до застосування.

#### Список використаних джерел:

1. TypeScript Handbook. TypeScript Documentation. 2024. URL: <https://www.typescriptlang.org/docs/handbook/intro.html> (Дата звернення: 12.08.2024)
2. Union Arrays in TypeScript. Catch TS. 2024. URL: <https://catchts.com/union-array> (Дата звернення: 22.08.2024)
3. Sitnikov D. Rough set based algebraic approach to modelling complex systems / Sitnikov D., Ryabov O., Mishcheriakov I., Kovalenko A. A. // The International Journal of Design & Nature and Ecodynamics. – Vol. 13, No. 3 (2018). – pp. 324–329. ISSN: 1755-7437 (paper format), ISSN: 1755-7445 (online), DOI: 10.2495 / DNE-V13-N3-324 329; (Scopus)