



Наукові перспективи  
Видавнича група

№ 1 (29)

2024

# НАУКА і ТЕХНІКА

серія: право, серія: економіка, серія: педагогіка,  
серія: техніка, серія: фізико-математичні науки

СЬОГОДНІ

З Україною

в серці!



**Видавнича група «Наукові перспективи»**

**Громадська наукова організація «Всеукраїнська Асамблея  
докторів наук із державного управління»**

**Громадська організація «Асоціація науковців України»**

# ***«Наука і техніка сьогодні»***

*(Серія «Педагогіка», Серія «Право», Серія «Економіка»,  
Серія «Фізико-математичні науки», Серія «Техніка»)*

**Випуск № 1(29) 2024**

**Київ – 2024**

**Publishing Group «Scientific Perspectives»**

**Public Scientific Organization «Ukrainian Assembly of  
Doctors of Sciences in Public Administration»**

**Public organization «Association of Scientists of Ukraine»**

***"Science and technology today"***  
*("Pedagogy" series, "Law" series, "Economics" series,  
"Physical and mathematical sciences" series, "Technics" series)*

**Issue № 1(29) 2024**

**Kiev – 2024**



**«Наука і техніка сьогодні» (Серія «Педагогіка», Серія «Право»,  
Серія «Економіка», Серія «Фізико-математичні науки», Серія «Техніка»):  
журнал. 2024. № 1(29) 2024. С. 831**



*Згідно наказу Міністерства освіти і науки України від 07.04.2022 №320 журналу присвоєно категорію "Б" із економіки та педагогіки (спеціальності – 015 - Педагогічні науки; 076 - Економічні науки)*

*Згідно наказу Міністерства освіти і науки України від 06.06.2022 № 530 журналу присвоєно категорію "Б" із права (спеціальність – 081 Юридичні науки)*

*Згідно наказу Міністерства освіти і науки України від 10.10.2022 № 894 журналу присвоєно категорію "Б" із техніки (спеціальність - 122 Комп'ютерні науки)*

*Журнал видається за підтримки Міждержавної гільдії інженерів консультантів, Інституту філософії та соціології Національної Академії Наук Азербайджану (Баку, Азербайджан), громадської організації «Християнська академія педагогічних наук України» та громадської організації «Всеукраїнська асоціація педагогів і психологів з духовно-морального виховання»*

*Рекомендовано до видавництва Президією громадської наукової організації «Всеукраїнська Асамблея докторів наук з державного управління» (Рішення від 24.01.2024, № 6/1-24)*



Журнал включено до міжнародної наукометричної бази Index Copernicus (IC), міжнародної пошукової системи Google Scholar та до міжнародної наукометричної бази даних Research Bible

**Головний редактор:** Сопілко Ірина Миколаївна - доктор юридичних наук, професор, Відмінник освіти України, Лауреат Премії Президента України для молодих вчених, Лауреат Премії Верховної Ради України найталановитішим молодим ученим в галузі фундаментальних і прикладних досліджень та науково-технічних розробок, академік Академії наук вищої школи України, Заслужений юрист України (Київ, Україна)

**Редакційна колегія:**

- Артемчук Володимир Олександрович - доктор технічних наук, старший науковий співробітник, старший науковий співробітник Інституту проблем моделювання в енергетиці ім. Г.Є. Пухова НАН України (Київ, Україна)
- Бахов Іван Степанович – доктор педагогічних наук, професор, завідувач кафедри іноземної філології та перекладу Міжрегіональної академії управління персоналом (Київ, Україна)
- Будник Вікторія Анатоліївна - кандидат економічних наук, професор, професор кафедри бізнес-логістики та транспортних технологій Державного університету інфраструктури та технологій (Київ, Україна)
- Волк Павло Павлович – доцент кафедри водної інженерії та водних технологій Національного університету водного господарства та природокористування (Рівне, Україна)
- Гирка Ольга Ігорівна - кандидат технічних наук, доцент, доцент кафедри товарознавства, митної справи та управління якістю Львівського торговельно-економічного університету (Львів, Україна)
- Гнатюк Сергій Олександрович - кандидат технічних наук, доцент, заступник декана факультету аеронавігації, електроніки та телекомунікацій Національного авіаційного університету (Київ, Україна)
- Дацій Олександр Іванович - доктор економічних наук, професор, Заслужений працівник освіти України, завідувач кафедри фінансів, банківської та страхової справи Міжрегіональної академії управління персоналом (Київ, Україна)
- Дівізніюк Михайло Михайлович - доктор фізико-математичних наук, професор, Завідувач відділу Відділу цивільного захисту та інноваційної діяльності Державної установи Інститут геохімії навколишнього середовища Національної академії наук України" (Київ, Україна)
- Дяденчук Альона Федорівна - кандидат технічних наук, старший викладач кафедри вищої математики і фізики Таврійського державного агротехнологічного університету імені Дмитра Моторного (Мелітополь, Україна)
- Забулонов Юрій Леонідович - доктор технічних наук, професор, Член-кореспондент НАН України, директор Державної установи «Інститут геохімії навколишнього середовища Національної академії наук України» (Київ, Україна)
- Ільїн Валерій Юрійович - доктор економічних наук, професор (Київ, Україна)
- Ільїна Анастасія Олександрівна - кандидат економічних наук, доцент, доцент кафедри публічного управління і адміністрування Національного торговельно-економічного університету (Київ, Україна)
- Кардаш Оксана Любомирівна – кандидат економічних наук, доцент кафедри комп'ютерних технологій та економічної кібернетики Навчально-наукового інституту автоматичної, кібернетики та обчислювальної техніки Національного університету водного господарства та природокористування (м. Рівне, Україна)
- Квасніков Володимир Павлович – доктор технічних наук, професор, завідувач кафедри комп'ютеризованих електротехнічних систем та технологій Національного авіаційного університету (Київ, Україна)

- Коваленко Валентин Васильович - доктор юридичних наук, професор, провідний науковий співробітник сектору авторського права та суміжних прав лабораторії авторського права та інформаційних технологій Науково-дослідного центру судової експертизи з питань інтелектуальної власності Міністерства юстиції України (Київ, Україна)
- Коваленко Олена Михайлівна - кандидат педагогічних наук, провідний науковий співробітник відділу профільного навчання Інституту педагогіки НАПН України (Київ, Україна)
- Комнатний Сергій Олександрович - докторант кафедри філософії права та юридичної логіки Національної академії внутрішніх справ (Київ, Україна)
- Кравчук Володимир Миколайович — доктор юридичних наук, доцент, доцент кафедри конституційного, адміністративного та міжнародного права Волинського національного університету імені Лесі Українки (Луцьк, Україна)
- Кузьмич Людмила Володимирівна - доктор технічних наук, головний науковий співробітник Інституту водних проблем і меліорації Національної академії аграрних наук України (Київ, Україна)
- Куницький Сергій Олегович - кандидат технічних наук, старший дослідник, провідний науковий співробітник науково-дослідної частини Національного університету водного господарства та природокористування (Рівне, Україна)
- Лук'янчук Олександр Петрович — кандидат технічних наук, доцент, доцент кафедри будівельних, дорожніх, меліоративних, сільськогосподарських машин та обладнання Національного університету водного господарства та природокористування (Рівне, Україна)
- Маджд Світлана Михайлівна - доктор технічних наук, професор, професор кафедри зеленої економіки та економіки природокористування Державної екологічної академії післядипломної освіти та управління (Київ, Україна)
- Мануель Давид Массено - доцент відділу права та захисту даних, старший науковий співробітник і член координаційного комітету лабораторії UbiNET, запрошений член PDPC, член-консультант комісії цифрового права муніципальних адвокатських колегій Кампінаса та Прая-Гранде (Сан-Паулу), а також Комісії з інновацій, управління та технологій муніципальної адвокатської колегії Гуарульуса, коментатор ЮДА, почесний член IDEIA Institute, член Наукового комітету MISCHR, член EDEN, член-кореспондент RedNAC, член UMAU, член-кореспондент UBAU (Португалія)
- Микитин Тарас Миронович - кандидат технічних наук, завідувач кафедри менеджменту Рівненського державного гуманітарного університету (Рівне, Україна)
- Миргород-Карпова Валерія Валеріївна - кандидат юридичних наук, заступник директора з наукової роботи, старший викладач кафедри адміністративного, господарського права та фінансово-економічної безпеки Сумського державного університету (Суми, Україна)
- Мізюк Вікторія Анатоліївна - кандидат педагогічних наук, доцент, декан факультету управління, адміністрування та інформаційної діяльності Ізмаїльського державного гуманітарного університету (Ізмаїл, Україна)
- Мірошніченко Валентина Іванівна - доктор педагогічних наук, професор, завідувач кафедри психології, педагогіки та соціально-економічних дисциплін Національної академії державної прикордонної служби України імені Богдана Хмельницького (Хмельницький, Україна)
- Міхальський Томаш — доктор наук, доцент кафедри географії регіонального розвитку Гданського університету (Польща)
- Огієнко Микола Миколайович - кандидат технічних наук, професор кафедри організації авіаційних робіт та послуг Національного авіаційного університету (Київ, Україна)
- Одарченко Роман Сергійович - завідувач кафедри телекомунікаційних та радіоелектронних систем Національного авіаційного університету (Київ, Україна)
- Оніщенко Наталія Миколаївна - доктор юридичних наук, професор, Заслужений юрист України, академік НАПН України, завідувач відділу теорії держави і права Інституту держави і права ім. В.М.Корецького НАН України (Київ, Україна)
- Опанасенко Володимир Миколайович — доцент кафедри комп'ютеризованих електротехнічних систем та технологій Національного авіаційного університету (Київ, Україна)
- Охрімченко (Жмурко) Тетяна Олександрівна - старший науковий співробітник кафедри комп'ютеризованих систем управління Національного авіаційного університету (Київ, Україна)
- Павлов Костянтин Володимирович — доктор економічних наук, професор, завідувач кафедри підприємництва і маркетингу Волинського національного університету імені Лесі Українки (Луцьк, Україна)
- Поліщук Віталій Васильович — кандидат сільськогосподарських наук, завідувач відділу зрошення, відділення меліорації Інституту водних проблем і меліорації Національної академії аграрних наук України (Київ, Україна)
- Приходькіна Наталія Олексіївна - доктор педагогічних наук, професор кафедри педагогіки, адміністрування і спеціальної освіти Навчально-наукового інституту менеджменту та психології ДЗВО «Університет менеджменту освіти» НАПН України (Київ, Україна)
- Стахова Анжеліка Петрівна — старший викладач кафедри комп'ютеризованих електротехнічних систем та технологій Національного авіаційного університету (Київ, Україна)
- Турчинова Ганна Володимирівна — кандидат педагогічних наук, доцент, декан факультету природничо-географічної освіти та екології Національного педагогічного університету імені М.П. Драгоманова (Київ, Україна)
- Фесенко Андрій Олексійович - кандидат технічних наук, асистент кафедри кібербезпеки та захисту інформації Київського національного університету імені Тараса Шевченка. (Київ, Україна)
- Черненко Варвара Петрівна - кандидат фізико-математичних наук, доцент кафедри інформатики і вищої математики Кременчуцького національного університету імені Михайла Остроградського (Кременчук, Україна)
- Чернуха Надія Миколаївна — доктор педагогічних наук, професор, професор кафедри соціальної реабілітації та соціальної педагогіки Київського національного університету імені Тараса Шевченка (Київ, Україна)
- Чумак Оксана Володимирівна - доктор економічних наук, доцент, науковий співробітник відділу статистики і аналітики вищої освіти Державної наукової установи «Інститут освітньої аналітики», (Київ, Україна)
- Шандра Наталія Андріївна - кандидат педагогічних наук, доцент кафедри іноземних мов для природничих факультетів Львівського національного університету імені Івана Франка (Львів, Україна)
- Шеремет Інеса Володимирівна - кандидат педагогічних наук, доцент, доцент кафедри медикобіологічних та валеологічних основ охорони життя і здоров'я Національного педагогічного університету ім. М. П. Драгоманова (Київ, Україна)
- Якимчук Аліна Юріївна - доктор економічних наук, професор, Академік економічних наук України, професор кафедри державного управління, документознавства та інформаційної діяльності Національного університету водного господарства та природокористування (Рівне, Україна)
- Якимчук Олег Феодосійович - керівник групи білінгу Відділу бізнес-систем Департаменту інформаційних технологій ПРАТ «Рівнеобленерго» (Рівне, Україна)
- Яцишин Андрій Васильович - доктор технічних наук, старший науковий співробітник, провідний науковий співробітник Відділу цивільного захисту та інноваційної діяльності Державної установи Інститут геохімії навколишнього середовища Національної академії наук України (Київ, Україна)

Статті розміщені в авторській редакції. Відповідальність за зміст та орфографію поданих матеріалів несуть автори.

© автори статей, 2024

© Видавнича група «Наукові перспективи», 2024

### СЕРІЯ «Техніка»

<b>Denysiuk V.O.</b> <i>REVIEW OF THE PARALLEL HYPERQUICK SORT ALGORITHM BY C#</i>	<b>605</b>
<b>Божко О.Ю.</b> <i>ВИКОРИСТАННЯ ВЕЛИКИХ МОВНИХ МОДЕЛЕЙ ДЛЯ РОЗПІЗНАВАННЯ ІНФОРМАЦІЇ В НУМІЗМАТИЧНИХ ОПИСАХ</i>	<b>615</b>
<b>Гирка О.І., Бодак М.П., Філь М.І., Рубай У.І.</b> <i>ПРОБЛЕМИ ЛОГІСТИКИ СОНЯШНИКОВОЇ ОЛІЇ В ПЕРІОД ВОЄННОГО СТАНУ ТА СУЧАСНІ ТЕНДЕНЦІЇ ЇХ ВИРІШЕННЯ</i>	<b>626</b>
<b>Горбачук В.М., Гавриленко С.О., Ніколенко Д.І.</b> <i>ЦИФРОВЕ СПІВРОБІТНИЦТВО З ЕКОНОМІЧНОЇ БЕЗПЕКИ</i>	<b>637</b>
<b>Дьогтева І.О., Нікіфорова Л.О., Шиян А.А., Жарінов С.С.</b> <i>МОДЕЛЬ ОПТИМІЗАЦІЇ ЕКСПЕРТИЗИ В КОНКУРСНОМУ ВІДБОРІ ПРОЄКТІВ ДОСЛІДЖЕНЬ І РОЗРОБОК</i>	<b>651</b>
<b>Іванов А.О., Кривонос О.М., Жуковський С.С.</b> <i>ОСОБЛИВОСТІ РЕАЛІЗАЦІЇ АЛГОРИТМУ ФОРЧУНА ДЛЯ ПОБУДОВИ ДІАГРАМИ ВОРОНОГО НА МОВІ ПРОГРАМУВАННЯ PYTHON</i>	<b>666</b>
<b>Карімов І.К., Карімов Г.І., Компанієць Р.А., Булай О.Ю.</b> <i>КОМП'ЮТЕРНО ОРІЄНТОВАНИЙ АЛГОРИТМ МОДЕЛЮВАННЯ МІСЦЕВОГО НАГРІВУ ПЛИТНИХ КОНСТРУКЦІЙ</i>	<b>689</b>
<b>Комаров В.О., Кокошинський В.В., Думітраш В.О., Яковчук О.В., Зінченко М.О., Лазута Р.Р., Безносенко С.Ю.</b> <i>ТЕХНОЛОГІЯ SDN ЯК ІННОВАЦІЙНИЙ РУШІЙ ДЛЯ МОДЕРНІЗАЦІЇ ВІЙСЬКОВОЇ ТРАНСПОРТНОЇ ТЕЛЕКОМУНІКАЦІЙНОЇ МЕРЕЖІ УКРАЇНИ</i>	<b>701</b>
<b>Кононихін О.С., Прачик В.А., Щепетільников С.Ю.</b> <i>МОДЕЛЬ ВИБОРУ МЕРЕЖЕВОГО ОБЛАДНАННЯ ЛОГІСТИЧНОГО ПІДПРИЄМСТВА В УМОВАХ ІНТЕРВАЛЬНОЇ ІНФОРМАЦІЇ</i>	<b>718</b>
<b>Корехов А.О.</b> <i>РОЗРОБКА СКЛАДОВИХ СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ НАЧАЛЬНИКОМ ВІДДІЛЕННЯ АВТОМОБІЛЬНОГО ТА БРОНЕТАНКОВОГО ЗАБЕЗПЕЧЕННЯ</i>	<b>728</b>
<b>Корнута В.А., Корнута О.В., Катамай Ю.В., Меренько Б.І.</b> <i>ОСОБЛИВОСТІ ПРОЄКТУВАННЯ ІНТЕРФЕЙСІВ КОРИСТУВАЧА ДЛЯ МОБІЛЬНИХ ДОДАТКІВ</i>	<b>738</b>



№ 1(29)  
2024

НАУКА  
і ТЕХНІКА

серії: право, економіка, педагогіка,  
техніка, фізико-математичні науки

СЬОГОДНІ



## СЕРІЯ «Техніка»

UDC 004.8

[https://doi.org/10.52058/2786-6025-2024-1\(29\)-605-614](https://doi.org/10.52058/2786-6025-2024-1(29)-605-614)

**Denysiuk Valerii Olexandrovich** PhD, assistant professor of Computer Sciences Department, Vinnytsia National Technical University, Khmelnytske shose, 95, Vinnytsia, 21021, tel.: (093) 632-58-58, <https://orcid.org/0000-0003-1057-3518>

### REVIEW OF THE PARALLEL HYPERQUICK SORT ALGORITHM BY C#

**Abstract.** Sorting is used to arrange large arrays of data and present them in a certain structured form. With the development of distributed systems and parallel computing, algorithms were also developed that used the advantages of these technologies to improve and increase the efficiency of the tasks of sorting data arrays. The article discusses the algorithm for parallel hyperquick sorting of data arrays by C# programming language. Hyperquicksort is a type of quicksort algorithm that uses multiple nested items to divide a list into sublists. This allows the algorithm to sort the list faster than the standard quicksort algorithm, The algorithm uses multiple nested elements to divide the list into more than two sublists, which allows the list to be sorted more efficiently, especially when the list is already partially sorted or when it has other special properties. Using multiple aggregates can allow the algorithm to skip large sections of the list that are already in the correct order, instead of recursively sorting them. In the case where the list has some other special property, such as a large number of elements that are equal to the reference part, using multiple reference elements can allow the algorithm to partition the list more efficiently.

The algorithm is an improvement using a simple combination of serial and parallel approaches. Reduces the problem of load balancing. Improves finding the median by sequentially sorting the sublists using a single reference point that is broadcast to all processes at the beginning of the algorithm.

The flow graph and program code of the parallel hyperquick sorting algorithm were developed by C# language.

The algorithm was tested for a large array of data. To test the parallel hyperquick sort algorithm, a program was created that implements the parallel hyperquick sort algorithm and related algorithms, namely: sequential quick sort;

605



naive parallel sorting; optimized parallel sorting; parallel-serial sorting; parallel quicksort by regular sampling. The testing program has the following capabilities: generation of an array with random values according to the size specified by the user; sorting a randomly generated array of data using one of the selected algorithms; measuring the time spent on sorting an array of N elements. The analysis of the obtained data confirms the good time performance of the proposed algorithm for parallel hyperquick sorting of data arrays in comparison with existing sorting algorithms.

**Keywords:** program code, sorting, algorithm, sorting sublist, parallel algorithm, hyperfast sorting

**Денисюк Валерій Олександрович** кандидат технічних наук, доцент, доцент кафедри комп'ютерних наук, Вінницький національний технічний університет, Хмельницьке шосе, 95, м. Вінниця, 21021, тел.: (093) 632-58-58, <https://orcid.org/0000-0003-1057-3518>

## РОЗГЛЯД АЛГОРИТМА ПАРАЛЕЛЬНОГО ГІПЕРШВИДКОГО СОРТУВАННЯ МОВОЮ C#

**Анотація.** Для впорядкування великих масивів даних та подання їх в певному структурованому вигляді використовується сортування. З розвитком розподілених систем та паралельних обчислень розвивалися і алгоритми, які використовували переваги даних технологій для вдосконалення і підвищення ефективності задач сортування масивів даних. У статті розглянуто алгоритм паралельного гіпершвидкого сортування масивів даних мовою програмування C#. Гіпершвидке сортування є різновидом алгоритму швидкого сортування, який використовує кілька зведених елементів для поділу списку на підсписки. Це дозволяє алгоритму сортувати список швидше, ніж стандартний алгоритм швидкого сортування, Алгоритм використовує кілька зведених елементів для поділу списку на більше ніж два підсписки, що дозволяє сортувати список ефективніше, особливо коли список вже частково відсортовано або коли він має інші спеціальні властивості. Використання кількох зведених елементів може дозволити алгоритму пропускати великі розділи списку, які вже знаходяться в правильному порядку, замість того, щоб їх рекурсивно сортувати. У випадку, коли список має деякі інші особливі властивості, наприклад, велика кількість елементів, які дорівнюють опорній частині, використання кількох опорних елементів може дозволити алгоритму розділити список більш ефективно.

Алгоритм є вдосконаленням використання простої комбінації послідовного та паралельного підходів. Зменшує проблему балансування навантаження. Покращує знаходження медіани шляхом послідовного сортування підсписків за допомогою однієї опорної точки, яка транслюється усім процесам на початку алгоритму.





Розроблено потоковий граф і програмний код алгоритму паралельного гіпершвидкого сортування мовою програмування C#.

Проведено тестування алгоритму для великого масиву даних. Для тестування алгоритму паралельного гіпершвидкого сортування створено програму, яка реалізує паралельний алгоритм гіпершвидкого сортування та споріднені алгоритми, а саме: послідовне швидке сортування; найвніше паралельне сортування; оптимізоване паралельне сортування; паралельно-послідовне сортування; паралельне швидке сортування шляхом регулярної вибірки. Програма тестування має такі можливості: генерування масиву з випадковими значеннями за розміром заданим користувачем; сортування випадково згенерованого масиву даних за допомогою одного із обраних алгоритмів; вимірювання часу витраченого на сортування масиву з N елементів. Аналіз отриманих даних підтверджує хороші часові показники запропонованого алгоритма паралельного гіпершвидкого сортування масивів даних у порівнянні з існуючими алгоритмами сортування.

**Ключові слова:** програмний код, сортування, алгоритм, підписок сортування, паралельний алгоритм, гіпершвидке сортування

**Introduction.** In the era of the development of digital technologies, the most important sciences remain the sciences of working with data: their arrangement, editing, search, organization, etc. Sorting is used to arrange large arrays of data and present them in a certain structured form.

Many different algorithms have been developed for data sorting, which have their own advantages and disadvantages [1,2].

With the development of distributed systems and parallel computing, algorithms were also developed that used the advantages of these technologies to improve and increase efficiency. One of these algorithms is the hyper-fast sorting algorithm. It is parallel by default because it is based on an advanced parallel-serial sort method. In turn, this algorithm consists of a parallel and sequential sorting method, according to the name. There is also an even more innovative technology of parallel sorting with regular sampling, but the efficiency of each of these algorithms will depend on many factors, namely: the size of the array, its filling, the power of the processor, the number of cores and threads.

### **The main part.**

**Analysis of the subject area.** Hyper-quicksort is a type of quicksort algorithm that uses multiple nested elements to divide a list into sublists. This allows the algorithm to sort the list faster than the standard quicksort algorithm, especially for lists that are already partially sorted or have some other special properties [3].

In a standard quicksort algorithm, one aggregate element is selected from the list, and the other elements are split into two sublists based on whether they are smaller than or greater than the aggregate. The anchor point is then placed in the final position and the two sublists are recursively sorted using the same process.



In contrast to the standard quicksort algorithm, a hyperquicksort algorithm is considered, which uses several composite elements to divide a list into more than two sublists. This allows the algorithm to sort the list more efficiently in certain cases, especially when the list is already partially sorted or when it has other special properties.

For example, if the list is already partially sorted, using multiple aggregates can allow the algorithm to skip large sections of the list that are already in the correct order, instead of recursively sorting them. In the case where the list has some other special property, such as a large number of elements that are equal to the reference part, using multiple reference elements can allow the algorithm to partition the list more efficiently.

In general, hyper-quicksort is a useful variant of the quicksort algorithm that can provide improved performance for certain types of lists. However, this is not always the best choice, and the standard quicksort algorithm may be more efficient in some cases.

One of the possible hyperquick sorting algorithms is as follows.

1. If the size of the list (sorting array) is smaller than some threshold value, for example, 8 elements, then it is better to use another sorting algorithm (insertion, merging, etc. [1, 2]).

2. Select several pivot elements from the list.

3. Split the input list into sublists based on the aggregate elements such that all elements in each sublist are less than or equal to the corresponding aggregate element.

4. Sort each sublist using the hyperquick sorting algorithm.

5. Merge the sorted sublists to get the final sorted list.

6. Return the sorted list.

It is proposed to investigate a parallel hyperquick sorting algorithm that uses the number of processes (threads) equal to the number of reference elements.

It is based on the modified parallel quicksort algorithm, which uses the sequential quicksort and optimized parallel quicksort approaches.

**Mathematical modeling of the parallel hyperquick sorting algorithm.** By default, hyper-quicksort is a parallel algorithm because it uses the sequential quicksort and parallel quicksort algorithms.

In turn, the quick sort algorithm is not parallel by default. There are several ways to make the quicksort algorithm parallel.

First, using a parallel divide-and-conquer strategy where each subtask is solved in parallel. This can be done using techniques such as multithreading or multiprocessing.

Second, using a parallel sort algorithm, such as parallel merge sort, as a variant of the merge sort algorithm. This can be done using methods such as:

- SIMD - one instruction, several data;
- GPGPU - general-purpose computing on graphics processors.



Third, the use of a hybrid approach. Namely: a quick sort algorithm is used to initially split the data, and then a parallel sort algorithm is used to sort the resulting sub-solutions. This will effectively balance the high performance benefits of fastsort in the average case with the parallelization capabilities of other sorting algorithms.

Fig. 1 shows a parallel sort flow graph.

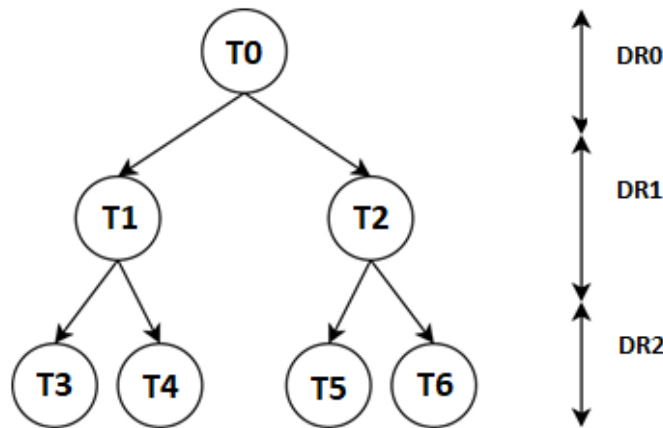


Fig. 1 Flow graph of parallel sorting (DR – Depth of Recursion, T – Tread)

Fig. 2 shows an example of a flow graph for hyperquick sorting of an array of 16 elements.

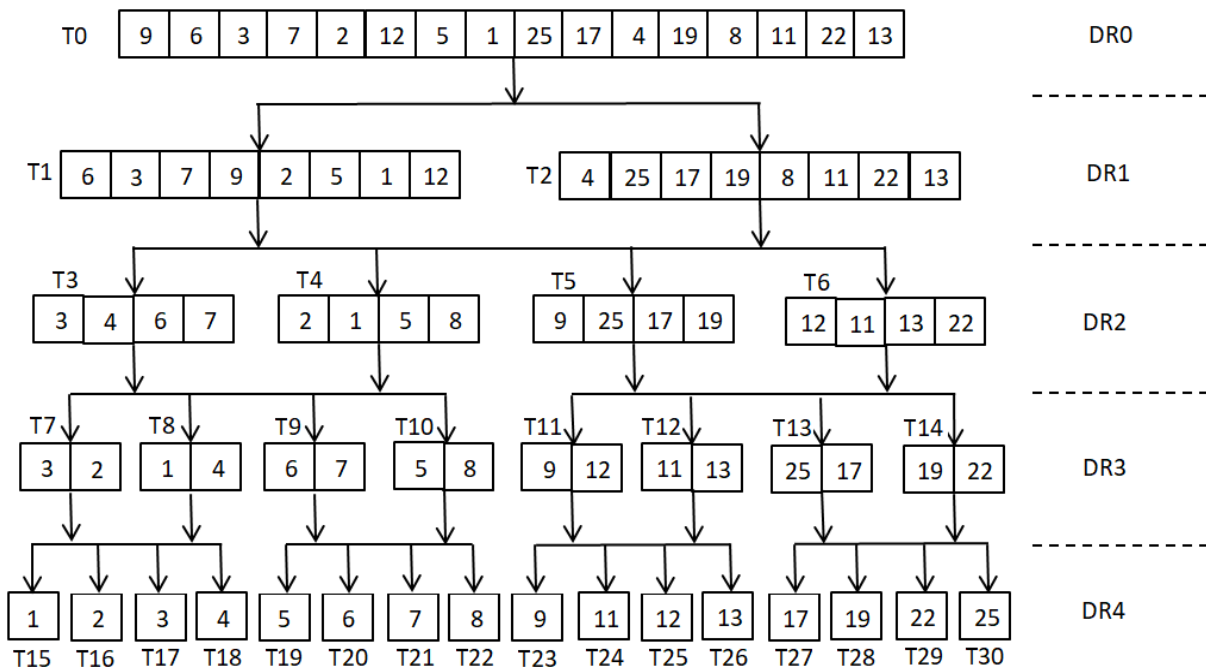


Fig. 2 A flow graph for hyperquick sorting of an array of 16 elements (DR – Depth of Recursion, T – Tread)



The parallel hyperquick sorting algorithm is as follows.

1. A list of size  $n$  is divided among  $n$  processes. Assume list of size 16 and 4 processes, each process will handle 4 elements.
2. A process among the four responsible for finding the pivot element, finds a pivot and broadcasts it to all processes which sort their sublists sequentially using the broadcasted pivot element. This step will improve chances of finding pivots close to the true median.
3. We perform the steps:
  - Randomly select a pivot element and broadcast it to all partner processes.
    - Each process will partition its elements and divide them into two groups according to the selected pivot.  $group1 \leq pivot \leq group2$ . This happens parallelly across all processes concurrently.
    - Each process in upper half of the process list sends its "low list" to a partner process in the lower half of the process list and it receives the "high list" in return.
  - 4. The remaining top half from one partner process and the received top half from the other partner process are merged into local sublist for each process.
  - 5. Recurse the upper half and lower half of each subprocess to achieve a sorted list.
  - 6. Finally merge the processes in order to get a fully sorted list.

**Software implementation of the algorithm.** Using  $n$  threads to sort  $n$  elements is not optimal. It is impossible to parallelize the process of dividing the array into subarrays under these conditions. In this case, it is best to use  $\log(n)$  processes to sort  $n$  elements in  $O(n)$  time.

The program that implements the parallel sorting algorithm Hyper Quick Sort is implemented as a console application using the C# [4, 5].

The `Parallel.Invoke()` method allows you to run several subtasks in parallel, but since the work implements recursive HyperQuickSort, you need to limit the depth of parallelization with this method. The best option is to use an additional variable of type integer, which will be equal to the number of GPU cores.

`Parallel.Invoke()` is a namespace method in the `System.Threading.Tasks` .NET framework that allows a set of specified methods to be executed simultaneously. It takes an array of delegate methods as input and returns when all methods have completed execution (Fig. 3).



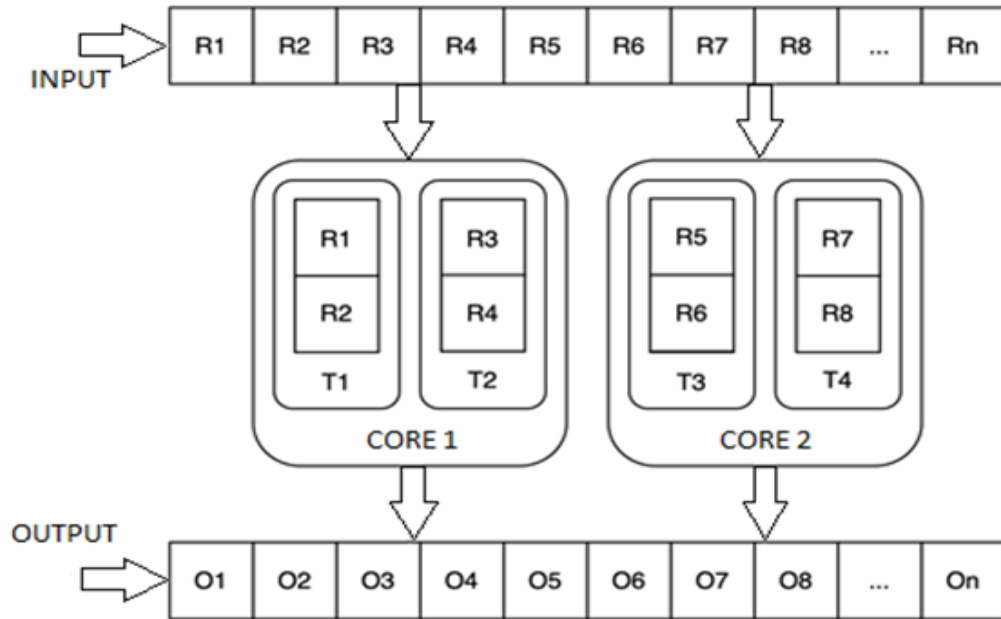
```
using System.Threading.Tasks;
namespace ConsoleApp
{
    class Program
    {
        static void Main(string[] args)
        {
            Parallel.Invoke(Method1, Method2, Method3);
        }
        static void Method1()
        {
            // method implementation
        }
        static void Method2()
        {
            // method implementation
        }
        static void Method3()
        {
            // method implementation
        }
    }
}
```

**Fig. 3** Example of *Parallel.Invoke()* use

In the example in Figure 3, the methods *Method1()*, *Method2()*, and *Method3()* will be executed simultaneously. *Parallel.Invoke()* returns when all methods have finished executing.

It is important to note that *Parallel.Invoke()* does not guarantee the order in which the methods will be executed. If you want to ensure that methods are executed in a specific order, you should use other synchronization methods, such as *Task.WaitAll()* or *Task.WhenAll()*.

The *Parallel.Invoke* method accepts an array of Action objects as a parameter, that is, it is possible to pass to this method a set of methods that will be called upon its execution, and the number of methods may be different. As with the *Task* class, it is possible to pass either a method name or a lambda expression. Fig. 4 shows the working diagram of the *Parallel.Invoke* method.



*Fig. 4 Operation scheme of the Parallel.Invoke method*

Therefore, if there are multiple cores on the target machine, these methods will be executed in parallel on different cores.

Fig. 5 shows an example of the parallel sorting algorithm Hyper Quick Sort implemented by C#.

**Testing the Hyper Quick Sort algorithm.** To test the parallel hyperquick sorting algorithm, a program was created that implements the parallel hyperquick sorting algorithm and related algorithms [1-3]: Sequential Quicksort; Naive Parallel Quicksort; Optimized Parallel Quicksort; Parallel Sequential Quicksort; Hyper QuickSort; Parallel Quicksort by Regular Sampling.

This program has the following features: generation of an array with random values according to the size specified by the user; sorting of a randomly generated array of data using one of the selected algorithms; measuring the time spent on sorting an array of N elements.

Table 1 provides information on the time to sort a randomly generated data array with the number of elements  $N = 1,000,000$ , elements of type integer in the range from 1 to 999.

Therefore, the analysis of the obtained data of table 1 confirms the good time performance of the parallel hyperquick sorting algorithm (the shortest time is 262 ms) in comparison with some existing sorting algorithms.

**Conclusions.** The investigated parallel hyperquick sort algorithm uses multiple nested elements to divide the list into more than two sublists, which allows the list to be sorted more efficiently, especially when the list is already partially sorted or when it has other special properties.





```
using System;
namespace hyperquicksort
{
    public class HyperQuickSort
    {
        public static void Sort(int[] data, int left, int right)
        {
            // Check if the data should be sorted in parallel if right - left > 1000
            {
                // Invoke two tasks to sort the left and right halves of
                // the array in parallel
                Parallel.Invoke( () => Sort(data, left, (left + right) / 2),
                                () => Sort(data, (left + right) / 2 + 1, right));
            }
            else
            { // Use a traditional quicksort algorithm
              // to sort the data
                Quicksort(data, left, right);
            }
        }

        static void Quicksort(int[] data, int left, int right)
        {
            // Check if the data is already sorted if (left
            // >= right)
            {
                return;
            }

            // Choose a pivot value and partition the data around it
            int pivot = data[(left + right) / 2];
            int i = left; int j =
            right; while (i < j)
            {
                while (data[i] < pivot)
                {
                    i++;
                }
                while (data[j] > pivot)
                {
                    j--;
                }
                if (i <= j)
                {
                    int temp = data[i]; data[i]
                    = data[j]; data[j] = temp;
                    i++;
                    j--;
                }
            }
            // Recursively sort the left and right halves of the data
            Quicksort(data, left, j);
            Quicksort(data, i, right);
        }
    }
}
```

*Fig. 5 Class HyperQuickSort*

**Table 1** Time to sort an array of 1,000,000 elements

Sort Type	Time (ms)
Sequential Quicksort	1 278
Naive Parallel Quicksort	950
Optimized Parallel Quicksort	924
Parallel Sequential Quicksort	1 310
Hyper QuickSort	262
Parallel Quicksort by Regular Sampling	30 931

The algorithm is an improvement using a simple combination of serial and parallel approaches. Improves the chances of finding the true median by sequentially sorting the sublists using a single reference point that is broadcast to all processes at the beginning of the algorithm. There is communication overhead when values are passed between partner processes. Reduces the problem of load balancing. Load imbalance can still occur, but the algorithm is better compared to existing ones.

The flow graph and program code of the parallel hyperquick sorting algorithm have been developed. When processing a data array of dimension  $m$ , it calculates  $\log(m)$  steps and  $n$  processes, the total time complexity is  $\Theta(n \log(m))$ . The complexity of the space is  $O(\log(m))$ . Testing confirmed the workability of the algorithm and good time characteristics. In future studies, it will be interesting to consider 3D quick sorting.

**References:**

1. D. E. Knuth, The Art of Computer Programming, Vol. 3: Sorting and Searching (3rd. ed.), Addison Wesley Longman Publishing Co., Inc., 1998. 812 p.
2. S. V. Koliadenko, V. O. Denysiuk, N. P. Yurchuk. Dyskretnyi analiz. Chastyna 1. Navchalnyi posibnyk. Vinnytsia: VNAU, 2019. 161s. [in Ukrainian].
3. Parallel Quick Sort. Retrieved from <https://iq.opengenus.org/parallel-quicksort/>
4. C# documentation. Retrieved from <https://learn.microsoft.com/en-us/dotnet/csharp/>
5. Task Class. Retrieved from <https://learn.microsoft.com/en-s/dotnet/api/system.threading.tasks.task?view =net-6.0>

**Література:**

1. D. E. Knuth, The Art of Computer Programming, Vol. 3: Sorting and Searching (3rd. ed.), Addison Wesley Longman Publishing Co., Inc., 1998. 812 p.
2. С. В. Коляденко, В. О. Денисюк, Н. П. Юрчук. Дискретний аналіз. Частина 1. Навчальний посібник. Вінниця: ВНАУ, 2019. 161с.
3. Parallel Quick Sort. <https://iq.opengenus.org/parallel-quicksort/>
4. C# documentation. <https://learn.microsoft.com/en-us/dotnet/csharp/>
5. Task Class. <https://learn.microsoft.com/en-s/dotnet/api/system.threading.tasks.task?view =net-6.0>



# Журнал

## **«Наука і техніка сьогодні»**

*(Серія «Педагогіка», Серія «Право», Серія «Економіка»,  
Серія «Фізико-математичні науки», Серія «Техніка»)*

**Випуск № 1(29) 2024**

Формат 60x90/8. Папір офсетний.  
Гарнітура Times New Roman.  
Ум. друк. арк. 8,2. Наклад 100 прим.

Видавець:

Громадська наукова організація «Всеукраїнська асамблея докторів наук з державного управління»  
Свідоцтво серія ДК №4957 від 18.08.2015 р., Андріївський узвіз, буд.11, оф 68, м. Київ, 04070.