

ОПЕРАЦІЙНІ СИСТЕМИ

Лабораторний проєкт 1

Алгоритми сортування

Кіщук Ярослав Ярославович

Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка
Факультет комп'ютерних наук та кібернетики

ОПЕРАЦІЙНІ СИСТЕМИ

Лабораторний проєкт 1

Алгоритми сортування

Виконав: студент групи _____ Кіщук Ярослав Ярославович

Перевірила: _____

1 Мета роботи

Опанувати реалізацію та експериментальне дослідження класичних і багатопотокових алгоритмів сортування масивів; порівняти їх за часом виконання, кількістю порівнянь і об'ємом додаткової пам'яті для різних розмірів даних та розподілів (найгірший / рівномірний / нормальний / найкращий випадок); перевірити відповідність вимірювань теоретичній складності та зробити висновки щодо доцільності вибору алгоритму залежно від умов.

2 Завдання

Реалізувати та дослідити алгоритми сортування масивів для типів `int` та `double`. Для кожного алгоритму оцінити:

- об'єм додаткової пам'яті V (байти);
- кількість порівнянь K ;
- час виконання T (мс).

Теоретично оцінити складність алгоритмів. Провести експерименти для розмірів $N \in \{10^4, 2 \cdot 10^4, 5 \cdot 10^4, 10^5, 2 \cdot 10^5, 5 \cdot 10^5, 10^6\}$ та чотирьох умов даних: найгірший випадок, рівномірний розподіл, нормальний розподіл, найкращий випадок. Результати подати у вигляді таблиць; за бажанням — графіків.

3 Реалізація підрахунку метрик

Метрики V , K і T збираються в одному проході експерименту так.

Кількість операцій K . Використовується тип-обгортка `Counted<T>` (`instrumented_type.hpp`) елементи масиву мають тип `Counted<int>` або `Counted<double>`. Кожне порівняння (оператори `<`, `<=`, `>`, `>=`, `==`, `!=`) та кожне присвоєння, копіювання або переміщення інкрементує thread-local лічильники (`InstrumentationStats`). Після виклику `sort()` зчитується сума порівнянь і присвоєнь (`getTotalOperations()`); це значення записується як K . Таким чином K відображає кількість елементарних операцій алгоритму над елементами.

Додаткова пам'ять V (байти). Під час збірки бенчмарку підключається модуль `memory_tracker.hpp`, який перевизначає глобальні `operator new[]` та `operator delete[]`. Кожна алокація додається до трекера (`MemoryTracker::add`), при звільненні віднімається (`subtract`). Перед кожним запуском сортування лічильник скидається; після `sort()` зчитується пікове значення `peak_bytes` — максимальний сумарний об'єм пам'яті, виділений одночасно під час сортування. Це і є V .

Час T (мс). Вимірюється за допомогою `std::chrono::steady_clock`: фіксується момент до виклику `strategy->sort()` і після нього; різниця переводиться в мілісекунди. Для умов “Рівномірний” та “Нормальний” виконується кілька прогонів (наприклад 100), і в таблицю записується середнє значення T ; для “Найгірший” та “Найкращий” — один прогон. Перед кожним прогоном генерується новий масив, скидаються трекер пам'яті та інструментація, щоб вимірювання не залежали від попередніх запусків.

4 Теоретична складність алгоритмів

- **InsertionSort** — $O(n^2)$; додаткова пам'ять $O(1)$.
- **QuickSort** — $O(n^2)$ у найгіршому, $O(n \log n)$ в середньому; додаткова пам'ять $O(\log n)$ (стека).
- **MergeSort** — $O(n \log n)$; додаткова пам'ять $O(n)$.
- **HeapSort** — $O(n \log n)$; додаткова пам'ять $O(1)$.
- **BubbleSort** — $O(n^2)$; додаткова пам'ять $O(1)$.
- **SelectionSort** — $O(n^2)$; додаткова пам'ять $O(1)$.
- **ShellSort** — залежить від послідовності кроків; типово $O(n^{1.25})$ – $O(n^2)$; додаткова пам'ять $O(1)$.
- **RadixSort** (для `int`) — $O(n \cdot k)$, де k — кількість розрядів; додаткова пам'ять $O(n + \text{розмір алфавіту})$.
- **CountingSort** (для `int`) — $O(n + \text{діапазон})$; додаткова пам'ять $O(\text{діапазон})$.
- **MultiThreadedInsertionSort** — та сама складність $O(n^2)$, паралелізація по блоках.
- **MultiThreadedQuickSort** — $O(n \log n)$ в середньому при паралелізації.
- **MultiThreadedMergeSort** — $O(n \log n)$; додаткова пам'ять $O(n)$.
- **MultiThreadedHeapSort** — $O(n \log n)$; додаткова пам'ять $O(1)$.
- **MultiThreadedBubbleSort** — $O(n^2)$; паралелізація по проходах.

5 Характеристики системи

Експерименти та вимірювання виконувалися на ноутбучі Apple MacBook Air з такими параметрами (за виводом Google Benchmark та збірки проєкту):

- **Модель:** Apple MacBook Air, чип Apple M4, архітектура ARM64.
- **Процесор:** 10 логічних ядер. Номінальна частота на macOS бенчмарком не зчитується (`hw.cpufrequency` недоступний); на вимірювання часу T це не впливає.
- **Кеш L1:** Data 64 KiB, Instruction 128 KiB.
- **Кеш L2:** Unified 4096 KiB (на кожне ядро).
- **ОС:** macOS (Darwin 25.2.0).
- **ОЗП:** 24 ГіБ. Для проведених розмірів масивів (до $N = 10^6$ елементів) достатня; на відносне порівняння алгоритмів за складністю та залежністю від N не впливає.

Абсолютні значення часу T залежать від конфігурації; на іншій машині вони можуть відрізнитися. Відносне порівняння алгоритмів за складністю та залежністю від N залишається коректним.

6 Таблиці з результатами експериментів

Експерименти проведено для типів `int` та `double`. Умови: Найгірший (Worst), Рівномірний (Uniform), Нормальний (Normal), Найкращий (Best). Для повільних алгоритмів частина вимірювань при великих N пропущена через таймаут (5 хв).

6.1 Тип `int`

`int`, $N = 10.0K$

Алгоритм	Умова	V (Б)	K	T (мс)	BigO
InsertionSort	Найгірший	0	100.01M	13.10	$0.13N^2$
InsertionSort	Рівномірний	0	49.98M	6.68	$0.07N^2$
InsertionSort	Нормальний	0	50.07M	6.59	$0.07N^2$
InsertionSort	Найкращий	0	30.00K	0.0050	$0.38N$
QuickSort	Найгірший	0	640.76K	0.1180	$0.89N \lg N$
QuickSort	Рівномірний	0	387.12K	0.3760	$2.77N \lg N$
QuickSort	Нормальний	0	382.45K	0.5930	$45.40N$
QuickSort	Найкращий	0	336.51K	0.0530	$0.39N \lg N$
MergeSort	Найгірший	534464	336.24K	0.4950	$50.15N$
MergeSort	Рівномірний	534464	387.68K	0.6730	$73.32N$
MergeSort	Нормальний	534464	387.69K	0.6750	$76.64N$
MergeSort	Найкращий	534464	331.84K	0.5130	$49.86N$
HeapSort	Найгірший	0	640.33K	0.3880	$3.20N \lg N$
HeapSort	Рівномірний	0	607.97K	0.4460	$3.82N \lg N$
HeapSort	Нормальний	0	607.81K	0.4590	$3.77N \lg N$
HeapSort	Найкращий	0	576.77K	0.4250	$3.29N \lg N$
BubbleSort	Найгірший	0	199.98M	13.00	$0.13N^2$
BubbleSort	Рівномірний	0	125.02M	25.50	$0.00N^3$
BubbleSort	Нормальний	0	124.99M	25.40	$0.00N^3$
BubbleSort	Найкращий	0	49.99M	12.90	$0.13N^2$
SelectionSort	Найгірший	0	50.02M	86.70	$0.67N^2$
SelectionSort	Рівномірний	0	50.02M	84.80	$0.64N^2$
SelectionSort	Нормальний	0	50.02M	87.00	$0.64N^2$
SelectionSort	Найкращий	0	50.02M	87.50	$0.61N^2$
ShellSort	Найгірший	0	475.15K	0.1040	$0.77N \lg N$
ShellSort	Рівномірний	0	651.99K	0.5740	$4.48N \lg N$
ShellSort	Нормальний	0	647.04K	0.5640	$4.46N \lg N$
ShellSort	Найкращий	0	360.01K	0.0630	$0.47N \lg N$
RadixSort	Найгірший	160000	170.0K	0.0640	$7.58N$
RadixSort	Рівномірний	200000	210.0K	0.0950	$0.64N \lg N$
RadixSort	Нормальний	200000	210.0K	0.1010	$0.66N \lg N$
RadixSort	Найкращий	160000	170.0K	0.0620	$7.38N$
CountingSort	Найгірший	80000	30.0K	0.0190	$1.79N$
CountingSort	Рівномірний	119992	30.0K	0.0250	$0.21N \lg N$
CountingSort	Нормальний	85316	30.0K	0.0210	$0.24N \lg N$
CountingSort	Найкращий	80000	30.0K	0.0190	$1.84N$
MultiThreadedInsertionSort	Найгірший	822080	0	561.00	$4.36N^2$
MultiThreadedInsertionSort	Рівномірний	822080	0	561.00	$4.14N^2$
MultiThreadedQuickSort	Найгірший	112	25.01K	0.0790	$12.55N$

(продовження)

Алгоритм	Умова	V (Б)	K	T (мс)	BigO
MultiThreadedQuickSort	Рівномірний	112	24.76K	0.2240	13.32N
MultiThreadedQuickSort	Нормальний	112	24.83K	0.2230	13.22N
MultiThreadedQuickSort	Найкращий	112	25.01K	0.0480	5.51N
MultiThreadedMergeSort	Найгірший	40080	25.0K	0.9730	85328.57lgN
MultiThreadedMergeSort	Рівномірний	40080	30.00K	0.6300	37.07N
MultiThreadedMergeSort	Нормальний	40080	30.00K	0.4080	28.71N
MultiThreadedMergeSort	Найкращий	40080	25.0K	0.3010	75623.36lgN
MultiThreadedHeapSort	Найгірший	0	640.33K	0.3680	3.05NlgN
MultiThreadedHeapSort	Рівномірний	0	607.90K	0.4280	3.62NlgN
MultiThreadedHeapSort	Нормальний	0	607.81K	0.4280	3.58NlgN
MultiThreadedHeapSort	Найкращий	0	576.77K	0.4060	3.55NlgN
MultiThreadedBubbleSort	Найгірший	0	199.98M	369.00	2.13N ²
MultiThreadedBubbleSort	Рівномірний	0	125.01M	206.00	1.71N ²
MultiThreadedBubbleSort	Нормальний	0	124.99M	145.00	1.84N ²
MultiThreadedBubbleSort	Найкращий	0	49.99M	12.60	0.13N ²

int, $N = 20.0K$

Алгоритм	Умова	V (Б)	K	T (мс)	BigO
InsertionSort	Найгірший	0	400.02M	53.00	0.13N ²
InsertionSort	Рівномірний	0	200.03M	26.40	0.07N ²
InsertionSort	Нормальний	0	199.96M	26.10	0.07N ²
InsertionSort	Найкращий	0	60.00K	0.0080	0.38N
QuickSort	Найгірший	0	1.41M	0.2570	0.89NlgN
QuickSort	Рівномірний	0	836.24K	0.8020	2.77NlgN
QuickSort	Нормальний	0	830.40K	1.38	45.40N
QuickSort	Найкращий	0	722.99K	0.1090	0.39NlgN
MergeSort	Найгірший	1148928	722.48K	1.01	50.15N
MergeSort	Рівномірний	1148928	835.38K	1.39	73.32N
MergeSort	Нормальний	1148928	835.37K	1.41	76.64N
MergeSort	Найкращий	1148928	713.68K	1.00	49.86N
HeapSort	Найгірший	0	1.38M	0.8690	3.20NlgN
HeapSort	Рівномірний	0	1.32M	0.9980	3.82NlgN
HeapSort	Нормальний	0	1.32M	1.01	3.77NlgN
HeapSort	Найкращий	0	1.26M	0.9190	3.29NlgN
BubbleSort	Найгірший	0	799.96M	52.00	0.13N ²
BubbleSort	Рівномірний	0	499.89M	103.00	0.00N ³
BubbleSort	Нормальний	0	499.76M	103.00	0.00N ³
BubbleSort	Найкращий	0	199.99M	51.40	0.13N ²
SelectionSort	Найгірший	0	200.05M	335.00	0.67N ²
SelectionSort	Рівномірний	0	200.05M	328.00	0.64N ²
SelectionSort	Нормальний	0	200.05M	330.00	0.64N ²
SelectionSort	Найкращий	0	200.05M	330.00	0.61N ²
ShellSort	Найгірший	0	1.03M	0.2220	0.77NlgN
ShellSort	Рівномірний	0	1.50M	1.25	4.48NlgN

(продовження)

Алгоритм	Умова	V (Б)	K	T (мс)	BigO
ShellSort	Нормальний	0	1.48M	1.23	$4.46N \lg N$
ShellSort	Найкращий	0	780.01K	0.1360	$0.47N \lg N$
RadixSort	Найгірший	400000	420.0K	0.1770	$7.58N$
RadixSort	Рівномірний	400000	420.0K	0.1670	$0.64N \lg N$
RadixSort	Нормальний	400000	420.0K	0.2030	$0.66N \lg N$
RadixSort	Найкращий	400000	420.0K	0.1720	$7.38N$
CountingSort	Найгірший	160000	60.0K	0.0390	$1.79N$
CountingSort	Рівномірний	239988	60.0K	0.0790	$0.21N \lg N$
CountingSort	Нормальний	173955	60.0K	0.0450	$0.24N \lg N$
CountingSort	Найкращий	160000	60.0K	0.0400	$1.84N$
MultiThreadedQuickSort	Найгірший	112	50.01K	0.1500	$12.55N$
MultiThreadedQuickSort	Рівномірний	112	51.07K	0.3610	$13.32N$
MultiThreadedQuickSort	Нормальний	112	49.28K	0.3550	$13.22N$
MultiThreadedQuickSort	Найкращий	112	50.01K	0.0760	$5.51N$
MultiThreadedMergeSort	Найгірший	80080	50.0K	0.5920	$85328.57 \lg N$
MultiThreadedMergeSort	Рівномірний	80080	60.00K	1.64	$37.07N$
MultiThreadedMergeSort	Нормальний	80080	60.00K	0.5100	$28.71N$
MultiThreadedMergeSort	Найкращий	80080	50.0K	0.3930	$75623.36 \lg N$
MultiThreadedHeapSort	Найгірший	0	1.38M	0.8280	$3.05N \lg N$
MultiThreadedHeapSort	Рівномірний	0	1.32M	0.9580	$3.62N \lg N$
MultiThreadedHeapSort	Нормальний	0	1.32M	0.9550	$3.58N \lg N$
MultiThreadedHeapSort	Найкращий	0	1.26M	0.8790	$3.55N \lg N$
MultiThreadedBubbleSort	Найгірший	0	799.96M	1528	$2.13N^2$
MultiThreadedBubbleSort	Рівномірний	0	499.79M	1008	$1.71N^2$

int, $N = 50.0K$

Алгоритм	Умова	V (Б)	K	T (мс)	BigO
InsertionSort	Найгірший	0	2500.05M	357.00	$0.13N^2$
InsertionSort	Рівномірний	0	1250.54M	164.00	$0.07N^2$
InsertionSort	Нормальний	0	1249.73M	163.00	$0.07N^2$
InsertionSort	Найкращий	0	150.00K	0.0200	$0.38N$
QuickSort	Найгірший	0	3.88M	0.6890	$0.89N \lg N$
QuickSort	Рівномірний	0	2.29M	2.17	$2.77N \lg N$
QuickSort	Нормальний	0	2.29M	2.13	$45.40N$
QuickSort	Найкращий	0	2.01M	0.3070	$0.39N \lg N$
MergeSort	Найгірший	3137856	1.97M	2.49	$50.15N$
MergeSort	Рівномірний	3137856	2.29M	3.59	$73.32N$
MergeSort	Нормальний	3137856	2.29M	3.85	$76.64N$
MergeSort	Найкращий	3137856	1.95M	2.49	$49.86N$
HeapSort	Найгірший	0	3.78M	2.43	$3.20N \lg N$
HeapSort	Рівномірний	0	3.62M	2.78	$3.82N \lg N$
HeapSort	Нормальний	0	3.62M	2.83	$3.77N \lg N$
HeapSort	Найкращий	0	3.46M	2.56	$3.29N \lg N$
BubbleSort	Найгірший	0	4999.90M	324.00	$0.13N^2$

(продовження)

Алгоритм	Умова	V (Б)	K	T (мс)	BigO
BubbleSort	Рівномірний	0	3123.97M	676.00	$0.00N^3$
BubbleSort	Нормальний	0	3124.39M	685.00	$0.00N^3$
BubbleSort	Найкращий	0	1249.97M	321.00	$0.13N^2$
SelectionSort	Найгірший	0	1250.12M	1889	$0.67N^2$
SelectionSort	Рівномірний	0	1250.12M	1864	$0.64N^2$
ShellSort	Найгірший	0	2.84M	0.6000	$0.77N \lg N$
ShellSort	Рівномірний	0	4.48M	3.44	$4.48N \lg N$
ShellSort	Нормальний	0	4.47M	3.42	$4.46N \lg N$
ShellSort	Найкращий	0	2.10M	0.3680	$0.47N \lg N$
RadixSort	Найгірший	1000000	1.05M	0.3840	$7.58N$
RadixSort	Рівномірний	1090000	1.14M	0.4670	$0.64N \lg N$
RadixSort	Нормальний	1000000	1.05M	0.4270	$0.66N \lg N$
RadixSort	Найкращий	1000000	1.05M	0.3730	$7.38N$
CountingSort	Найгірший	400000	150.0K	0.0900	$1.79N$
CountingSort	Рівномірний	599993	150.0K	0.1580	$0.21N \lg N$
CountingSort	Нормальний	442439	150.0K	0.1280	$0.24N \lg N$
CountingSort	Найкращий	400000	150.0K	0.0930	$1.84N$
MultiThreadedQuickSort	Найгірший	112	125.01K	0.5780	$12.55N$
MultiThreadedQuickSort	Рівномірний	112	126.24K	0.7160	$13.32N$
MultiThreadedQuickSort	Нормальний	112	128.24K	0.7120	$13.22N$
MultiThreadedQuickSort	Найкращий	112	125.01K	0.2720	$5.51N$
MultiThreadedMergeSort	Найгірший	200080	125.0K	1.21	$85328.57 \lg N$
MultiThreadedMergeSort	Рівномірний	200080	150.00K	3.04	$37.07N$
MultiThreadedMergeSort	Нормальний	200080	150.00K	1.49	$28.71N$
MultiThreadedMergeSort	Найкращий	200080	125.0K	1.27	$75623.36 \lg N$
MultiThreadedHeapSort	Найгірший	0	3.78M	2.31	$3.05N \lg N$
MultiThreadedHeapSort	Рівномірний	0	3.62M	2.71	$3.62N \lg N$
MultiThreadedHeapSort	Нормальний	0	3.62M	2.70	$3.58N \lg N$
MultiThreadedHeapSort	Найкращий	0	3.46M	2.45	$3.55N \lg N$

int, $N = 100.0K$

Алгоритм	Умова	V (Б)	K	T (мс)	BigO
InsertionSort	Найгірший	0	10000.10M	1337	$0.13N^2$
InsertionSort	Рівномірний	0	5001.12M	656.00	$0.07N^2$
InsertionSort	Нормальний	0	4999.43M	653.00	$0.07N^2$
InsertionSort	Найкращий	0	300.00K	0.0370	$0.38N$
QuickSort	Найгірший	0	8.42M	1.49	$0.89N \lg N$
QuickSort	Рівномірний	0	4.89M	4.60	$2.77N \lg N$
QuickSort	Нормальний	0	4.88M	4.50	$45.40N$
QuickSort	Найкращий	0	4.27M	0.6390	$0.39N \lg N$
MergeSort	Найгірший	6675712	4.19M	5.02	$50.15N$
MergeSort	Рівномірний	6675712	4.87M	7.39	$73.32N$
MergeSort	Нормальний	6675712	4.87M	7.69	$76.64N$
MergeSort	Найкращий	6675712	4.15M	4.99	$49.86N$

(продовження)

Алгоритм	Умова	V (Б)	K	T (мс)	BigO
HeapSort	Найгірший	0	8.07M	5.35	$3.20N\lg N$
HeapSort	Рівномірний	0	7.74M	6.46	$3.82N\lg N$
HeapSort	Нормальний	0	7.74M	6.33	$3.77N\lg N$
HeapSort	Найкращий	0	7.42M	5.47	$3.29N\lg N$
BubbleSort	Найгірший	0	19999.80M	1292	$0.13N^2$
BubbleSort	Рівномірний	0	12502.77M	4120	$0.00N^3$
ShellSort	Найгірший	0	6.09M	1.28	$0.77N\lg N$
ShellSort	Рівномірний	0	10.21M	7.47	$4.48N\lg N$
ShellSort	Нормальний	0	10.15M	7.44	$4.46N\lg N$
ShellSort	Найкращий	0	4.50M	0.7780	$0.47N\lg N$
RadixSort	Найгірший	2000000	2.10M	0.7510	$7.58N$
RadixSort	Рівномірний	2400000	2.50M	1.09	$0.64N\lg N$
RadixSort	Нормальний	2400000	2.50M	1.14	$0.66N\lg N$
RadixSort	Найкращий	2000000	2.10M	0.7320	$7.38N$
CountingSort	Найгірший	800000	300.0K	0.1780	$1.79N$
CountingSort	Рівномірний	1199992	300.0K	0.3540	$0.21N\lg N$
CountingSort	Нормальний	895665	300.0K	0.4300	$0.24N\lg N$
CountingSort	Найкращий	800000	300.0K	0.1830	$1.84N$
MultiThreadedQuickSort	Найгірший	112	250.01K	1.30	$12.55N$
MultiThreadedQuickSort	Рівномірний	112	250.87K	1.28	$13.32N$
MultiThreadedQuickSort	Нормальний	112	254.60K	1.27	$13.22N$
MultiThreadedQuickSort	Найкращий	112	250.01K	0.5600	$5.51N$
MultiThreadedMergeSort	Найгірший	400080	250.0K	2.20	$85328.57\lg N$
MultiThreadedMergeSort	Рівномірний	400080	300.00K	2.91	$37.07N$
MultiThreadedMergeSort	Нормальний	400080	300.00K	2.84	$28.71N$
MultiThreadedMergeSort	Найкращий	400080	250.0K	2.33	$75623.36\lg N$
MultiThreadedHeapSort	Найгірший	0	8.07M	5.11	$3.05N\lg N$
MultiThreadedHeapSort	Рівномірний	0	7.74M	6.08	$3.62N\lg N$
MultiThreadedHeapSort	Нормальний	0	7.74M	6.01	$3.58N\lg N$
MultiThreadedHeapSort	Найкращий	0	7.42M	6.07	$3.55N\lg N$

int, $N = 200.0K$

Алгоритм	Умова	V (Б)	K	T (мс)	BigO
InsertionSort	Найгірший	0	40000.20M	5171	$0.13N^2$
InsertionSort	Рівномірний	0	19998.26M	2596	$0.07N^2$
QuickSort	Найгірший	0	18.16M	3.06	$0.89N\lg N$
QuickSort	Рівномірний	0	10.35M	9.36	$2.77N\lg N$
QuickSort	Нормальний	0	10.32M	142.52	$45.40N$
QuickSort	Найкращий	0	9.04M	1.32	$0.39N\lg N$
MergeSort	Найгірший	14151424	8.88M	16.64	$50.15N$
MergeSort	Рівномірний	14151424	10.35M	29.59	$73.32N$
MergeSort	Нормальний	14151424	10.35M	30.12	$76.64N$
MergeSort	Найкращий	14151424	8.81M	17.00	$49.86N$
HeapSort	Найгірший	0	17.12M	20.87	$3.20N\lg N$

(продовження)

Алгоритм	Умова	V (Б)	К	T (мс)	BigO
HeapSort	Рівномірний	0	16.49M	22.79	$3.82N \lg N$
HeapSort	Нормальний	0	16.48M	22.43	$3.77N \lg N$
HeapSort	Найкращий	0	15.84M	17.83	$3.29N \lg N$
ShellSort	Найгірший	0	12.98M	3.01	$0.77N \lg N$
ShellSort	Рівномірний	0	23.29M	17.83	$4.48N \lg N$
ShellSort	Нормальний	0	23.15M	19.42	$4.46N \lg N$
ShellSort	Найкращий	0	9.60M	1.94	$0.47N \lg N$
RadixSort	Найгірший	4800000	5.0M	2.15	$7.58N$
RadixSort	Рівномірний	4800000	5.0M	2.09	$0.64N \lg N$
RadixSort	Нормальний	4800000	5.0M	2.50	$0.66N \lg N$
RadixSort	Найкращий	4800000	5.0M	2.10	$7.38N$
CountingSort	Найгірший	1600000	600.0K	0.4734	$1.79N$
CountingSort	Рівномірний	2399991	600.0K	2.12	$0.21N \lg N$
CountingSort	Нормальний	1809615	600.0K	1.25	$0.24N \lg N$
CountingSort	Найкращий	1600000	600.0K	0.6880	$1.84N$
MultiThreadedQuickSort	Найгірший	112	500.01K	3.29	$12.55N$
MultiThreadedQuickSort	Рівномірний	112	485.45K	7.72	$13.32N$
MultiThreadedQuickSort	Нормальний	112	513.16K	3.14	$13.22N$
MultiThreadedQuickSort	Найкращий	112	500.01K	1.42	$5.51N$
MultiThreadedMergeSort	Найгірший	800080	500.0K	5.65	$85328.57 \lg N$
MultiThreadedMergeSort	Рівномірний	800080	600.00K	7.01	$37.07N$
MultiThreadedMergeSort	Нормальний	800080	600.00K	6.84	$28.71N$
MultiThreadedMergeSort	Найкращий	800080	500.0K	5.44	$75623.36 \lg N$
MultiThreadedHeapSort	Найгірший	0	17.12M	10.48	$3.05N \lg N$
MultiThreadedHeapSort	Рівномірний	0	16.49M	12.36	$3.62N \lg N$
MultiThreadedHeapSort	Нормальний	0	16.48M	12.28	$3.58N \lg N$
MultiThreadedHeapSort	Найкращий	0	15.84M	10.11	$3.55N \lg N$

int, $N = 500.0K$

Алгоритм	Умова	V (Б)	К	T (мс)	BigO
QuickSort	Найгірший	0	49.91M	7.91	$0.89N \lg N$
QuickSort	Рівномірний	0	27.72M	23.81	$2.77N \lg N$
QuickSort	Нормальний	0	27.89M	30.00	$45.40N$
QuickSort	Найкращий	0	23.67M	5.97	$0.39N \lg N$
MergeSort	Найгірший	37902848	23.73M	41.04	$50.15N$
MergeSort	Рівномірний	37902848	27.79M	74.85	$73.32N$
MergeSort	Нормальний	37902848	27.79M	17967	$76.64N$
HeapSort	Найгірший	0	45.90M	50.75	$3.20N \lg N$
HeapSort	Рівномірний	0	44.47M	65.28	$3.82N \lg N$
HeapSort	Нормальний	0	44.44M	63.45	$3.77N \lg N$
HeapSort	Найкращий	0	42.98M	43.81	$3.29N \lg N$
ShellSort	Найгірший	0	33.86M	8.53	$0.77N \lg N$
ShellSort	Рівномірний	0	66.28M	53.94	$4.48N \lg N$
ShellSort	Нормальний	0	66.01M	46.70	$4.46N \lg N$

(продовження)

Алгоритм	Умова	V (Б)	K	T (мс)	BigO
ShellSort	Найкращий	0	25.50M	4.61	$0.47N \lg N$
RadixSort	Найгірший	12000000	12.50M	4.69	$7.58N$
RadixSort	Рівномірний	12680000	13.18M	5.70	$0.64N \lg N$
RadixSort	Нормальний	12000000	12.50M	5.71	$0.66N \lg N$
RadixSort	Найкращий	12000000	12.50M	4.89	$7.38N$
CountingSort	Найгірший	4000000	1.50M	1.97	$1.79N$
CountingSort	Рівномірний	5999990	1.50M	3.62	$0.21N \lg N$
CountingSort	Нормальний	4578449	1.50M	2.70	$0.24N \lg N$
CountingSort	Найкращий	4000000	1.50M	1.22	$1.84N$
MultiThreadedQuickSort	Найгірший	112	1.25M	7.00	$12.55N$
MultiThreadedQuickSort	Рівномірний	112	1.26M	6.24	$13.32N$
MultiThreadedQuickSort	Нормальний	112	1.21M	6.13	$13.22N$
MultiThreadedQuickSort	Найкращий	112	1.25M	2.63	$5.51N$
MultiThreadedMergeSort	Найгірший	2000080	1.25M	12.57	$85328.57 \lg N$
MultiThreadedMergeSort	Рівномірний	2000080	1.50M	15.56	$37.07N$
MultiThreadedMergeSort	Нормальний	2000080	1.50M	14.17	$28.71N$
MultiThreadedMergeSort	Найкращий	2000080	1.25M	13.04	$75623.36 \lg N$
MultiThreadedHeapSort	Найгірший	0	45.90M	27.83	$3.05N \lg N$
MultiThreadedHeapSort	Рівномірний	0	44.47M	34.93	$3.62N \lg N$
MultiThreadedHeapSort	Нормальний	0	44.44M	39.03	$3.58N \lg N$
MultiThreadedHeapSort	Найкращий	0	42.98M	27.00	$3.55N \lg N$

int, $N = 1.0M$

Алгоритм	Умова	V (Б)	K	T (мс)	BigO
QuickSort	Найгірший	0	106.44M	29.99	$0.89N \lg N$
QuickSort	Рівномірний	0	58.63M	67.02	$2.77N \lg N$
QuickSort	Нормальний	0	59.08M	64.48	$45.40N$
QuickSort	Найкращий	0	49.85M	8.85	$0.39N \lg N$
HeapSort	Найгірший	0	97.06M	95.90	$3.20N \lg N$
HeapSort	Рівномірний	0	93.94M	124.60	$3.82N \lg N$
HeapSort	Нормальний	0	93.88M	130.07	$3.77N \lg N$
HeapSort	Найкращий	0	91.00M	105.11	$3.29N \lg N$
ShellSort	Найгірший	0	71.72M	16.21	$0.77N \lg N$
ShellSort	Рівномірний	0	152.69M	103.76	$4.48N \lg N$
ShellSort	Нормальний	0	152.41M	103.87	$4.46N \lg N$
ShellSort	Найкращий	0	54.00M	9.80	$0.47N \lg N$
RadixSort	Найгірший	24000000	25.0M	10.65	$7.58N$
RadixSort	Рівномірний	28000000	29.0M	13.91	$0.64N \lg N$
RadixSort	Нормальний	28000000	29.0M	14.37	$0.66N \lg N$
CountingSort	Найгірший	8000000	3.0M	2.44	$1.79N$
CountingSort	Рівномірний	11999992	3.0M	4.17	$0.21N \lg N$
CountingSort	Нормальний	9226813	3.0M	3.39	$0.24N \lg N$
CountingSort	Найкращий	8000000	3.0M	1.89	$1.84N$
MultiThreadedQuickSort	Найгірший	112	2.50M	17.43	$12.55N$

(продовження)

Алгоритм	Умова	V (Б)	K	T (мс)	BigO
MultiThreadedQuickSort	Рівномірний	112	2.51M	12.56	$13.32N$
MultiThreadedQuickSort	Нормальний	112	2.53M	12.17	$13.22N$
MultiThreadedQuickSort	Найкращий	112	2.50M	5.88	$5.51N$
MultiThreadedMergeSort	Найгірший	4000080	2.50M	26.35	$85328.57lgN$
MultiThreadedMergeSort	Рівномірний	4000080	3.00M	30.88	$37.07N$
MultiThreadedMergeSort	Нормальний	4000080	3.00M	29.14	$28.71N$
MultiThreadedMergeSort	Найкращий	4000080	2.50M	21.26	$75623.36lgN$
MultiThreadedHeapSort	Найгірший	0	97.06M	58.06	$3.05NlgN$
MultiThreadedHeapSort	Рівномірний	0	93.94M	78.16	$3.62NlgN$
MultiThreadedHeapSort	Нормальний	0	93.88M	2624	$3.58NlgN$
MultiThreadedHeapSort	Найкращий	0	91.00M	57.72	$3.55NlgN$

6.2 Тип double

Для `double` не використовуються RadixSort та CountingSort (призначені для цілих).

`double`, $N = 10.0K$

Алгоритм	Умова	V (Б)	K	T (мс)
InsertionSort	Найгірший	0	100.01M	14.69
InsertionSort	Рівномірний	0	50.03M	6.95
InsertionSort	Нормальний	0	50.02M	6.94
InsertionSort	Найкращий	0	30.00K	0.0067
QuickSort	Найгірший	0	640.76K	0.1508
QuickSort	Рівномірний	0	391.79K	0.4083
QuickSort	Нормальний	0	394.03K	0.4133
QuickSort	Найкращий	0	336.51K	0.0549
MergeSort	Найгірший	1068928	336.24K	0.6030
MergeSort	Рівномірний	1068928	387.68K	0.9403
MergeSort	Нормальний	1068928	387.69K	0.9655
MergeSort	Найкращий	1068928	331.84K	0.5715
HeapSort	Найгірший	0	640.33K	0.8332
HeapSort	Рівномірний	0	607.96K	0.7665
HeapSort	Нормальний	0	607.99K	0.7262
HeapSort	Найкращий	0	576.77K	0.6353
BubbleSort	Найгірший	0	199.98M	15.96
BubbleSort	Рівномірний	0	124.92M	1294
BubbleSort	Нормальний	0	125.02M	28.22
BubbleSort	Найкращий	0	49.99M	13.53
SelectionSort	Найгірший	0	50.02M	177.53
SelectionSort	Рівномірний	0	50.02M	140.36
SelectionSort	Нормальний	0	50.02M	127.53
SelectionSort	Найкращий	0	50.02M	127.25
ShellSort	Найгірший	0	475.15K	0.1447

(продовження)

Алгоритм	Умова	V (Б)	K	T (мс)
ShellSort	Рівномірний	0	658.39K	0.6809
ShellSort	Нормальний	0	658.01K	0.6797
ShellSort	Найкращий	0	360.01K	0.0647
MultiThreadedInsertionSort	Найгірший	822080	0	599.38
MultiThreadedInsertionSort	Рівномірний	822080	0	23114
MultiThreadedQuickSort	Найгірший	112	25.01K	0.1212
MultiThreadedQuickSort	Рівномірний	112	23.76K	0.2737
MultiThreadedQuickSort	Нормальний	112	24.08K	0.2731
MultiThreadedQuickSort	Найкращий	112	25.01K	0.0521
MultiThreadedMergeSort	Найгірший	80080	25.0K	0.6927
MultiThreadedMergeSort	Рівномірний	80080	30.00K	0.9708
MultiThreadedMergeSort	Нормальний	80080	30.00K	0.9728
MultiThreadedMergeSort	Найкращий	80080	25.0K	0.6365
MultiThreadedHeapSort	Найгірший	0	640.33K	1.33
MultiThreadedHeapSort	Рівномірний	0	607.95K	1.22
MultiThreadedHeapSort	Нормальний	0	607.98K	1.11
MultiThreadedHeapSort	Найкращий	0	576.77K	0.9973
MultiThreadedBubbleSort	Найгірший	0	199.98M	212.66
MultiThreadedBubbleSort	Рівномірний	0	124.95M	5658

double, $N = 20.0K$

Алгоритм	Умова	V (Б)	K	T (мс)
InsertionSort	Найгірший	0	400.02M	52.21
InsertionSort	Рівномірний	0	200.11M	33.45
InsertionSort	Нормальний	0	200.05M	26.45
InsertionSort	Найкращий	0	60.00K	0.0107
QuickSort	Найгірший	0	1.41M	0.2803
QuickSort	Рівномірний	0	837.14K	0.8909
QuickSort	Нормальний	0	843.85K	0.8592
QuickSort	Найкращий	0	722.99K	0.1080
MergeSort	Найгірший	2297856	722.48K	1.15
MergeSort	Рівномірний	2297856	835.35K	1.95
MergeSort	Нормальний	2297856	835.36K	2.01
MergeSort	Найкращий	2297856	713.68K	1.17
HeapSort	Найгірший	0	1.38M	1.42
HeapSort	Рівномірний	0	1.32M	1.60
HeapSort	Нормальний	0	1.32M	1.66
HeapSort	Найкращий	0	1.26M	1.43
BubbleSort	Найгірший	0	799.96M	69.69
BubbleSort	Рівномірний	0	499.72M	113.56
BubbleSort	Нормальний	0	499.87M	115.18
BubbleSort	Найкращий	0	199.99M	56.25
SelectionSort	Найгірший	0	200.05M	493.90
SelectionSort	Рівномірний	0	200.05M	1349

(продовження)

Алгоритм	Умова	V (Б)	K	T (мс)
SelectionSort	Нормальний	0	200.05M	29449
ShellSort	Найгірший	0	1.03M	0.2329
ShellSort	Рівномірний	0	1.51M	1.48
ShellSort	Нормальний	0	1.51M	1.42
ShellSort	Найкращий	0	780.01K	0.1455
MultiThreadedQuickSort	Найгірший	112	50.01K	0.1833
MultiThreadedQuickSort	Рівномірний	112	48.30K	0.4609
MultiThreadedQuickSort	Нормальний	112	51.12K	0.5131
MultiThreadedQuickSort	Найкращий	112	50.01K	0.1095
MultiThreadedMergeSort	Найгірший	160080	50.0K	0.8341
MultiThreadedMergeSort	Рівномірний	160080	60.00K	1.23
MultiThreadedMergeSort	Нормальний	160080	60.00K	1.23
MultiThreadedMergeSort	Найкращий	160080	50.0K	1.14
MultiThreadedHeapSort	Найгірший	0	1.38M	2.19
MultiThreadedHeapSort	Рівномірний	0	1.32M	2.12
MultiThreadedHeapSort	Нормальний	0	1.32M	1.85
MultiThreadedHeapSort	Найкращий	0	1.26M	1.50

double, $N = 50.0K$

Алгоритм	Умова	V (Б)	K	T (мс)
InsertionSort	Найгірший	0	2500.05M	331.95
InsertionSort	Рівномірний	0	1249.98M	164.21
InsertionSort	Нормальний	0	1250.40M	164.29
InsertionSort	Найкращий	0	150.00K	0.0292
QuickSort	Найгірший	0	3.88M	0.7327
QuickSort	Рівномірний	0	2.29M	2.36
QuickSort	Нормальний	0	2.30M	2.39
QuickSort	Найкращий	0	2.01M	0.3049
MergeSort	Найгірший	6275712	1.97M	2.93
MergeSort	Рівномірний	6275712	2.29M	5.48
MergeSort	Нормальний	6275712	2.29M	5.76
MergeSort	Найкращий	6275712	1.95M	3.26
HeapSort	Найгірший	0	3.78M	4.08
HeapSort	Рівномірний	0	3.62M	6.18
HeapSort	Нормальний	0	3.62M	7.97
HeapSort	Найкращий	0	3.46M	5.89
BubbleSort	Найгірший	0	4999.90M	484.32
BubbleSort	Рівномірний	0	3125.59M	27005
ShellSort	Найгірший	0	2.84M	0.5528
ShellSort	Рівномірний	0	4.51M	3.94
ShellSort	Нормальний	0	4.50M	4.03
ShellSort	Найкращий	0	2.10M	0.3348
MultiThreadedQuickSort	Найгірший	112	125.01K	0.6937
MultiThreadedQuickSort	Рівномірний	112	125.44K	1.23

(продовження)

Алгоритм	Умова	V (Б)	К	T (мс)
MultiThreadedQuickSort	Нормальний	112	121.42K	1.19
MultiThreadedQuickSort	Найкращий	112	125.01K	0.4210
MultiThreadedMergeSort	Найгірший	400080	125.0K	2.73
MultiThreadedMergeSort	Рівномірний	400080	150.00K	2.86
MultiThreadedMergeSort	Нормальний	400080	150.00K	2.83
MultiThreadedMergeSort	Найкращий	400080	125.0K	1.80
MultiThreadedHeapSort	Найгірший	0	3.78M	4.30
MultiThreadedHeapSort	Рівномірний	0	3.62M	4.71
MultiThreadedHeapSort	Нормальний	0	3.62M	4.29
MultiThreadedHeapSort	Найкращий	0	3.46M	3.42

double, $N = 100.0K$

Алгоритм	Умова	V (Б)	К	T (мс)
InsertionSort	Найгірший	0	10000.10M	1321
InsertionSort	Рівномірний	0	4997.28M	12610
QuickSort	Найгірший	0	8.42M	1.57
QuickSort	Рівномірний	0	4.91M	4.93
QuickSort	Нормальний	0	4.89M	4.94
QuickSort	Найкращий	0	4.27M	0.6027
MergeSort	Найгірший	13351424	4.19M	6.31
MergeSort	Рівномірний	13351424	4.87M	12.10
MergeSort	Нормальний	13351424	4.87M	12.15
MergeSort	Найкращий	13351424	4.15M	6.68
HeapSort	Найгірший	0	8.07M	12.92
HeapSort	Рівномірний	0	7.74M	12.68
HeapSort	Нормальний	0	7.74M	10.35
HeapSort	Найкращий	0	7.42M	8.24
ShellSort	Найгірший	0	6.09M	1.22
ShellSort	Рівномірний	0	10.23M	8.26
ShellSort	Нормальний	0	10.22M	8.17
ShellSort	Найкращий	0	4.50M	0.6910
MultiThreadedQuickSort	Найгірший	112	250.01K	2.08
MultiThreadedQuickSort	Рівномірний	112	255.71K	2.25
MultiThreadedQuickSort	Нормальний	112	249.50K	2.57
MultiThreadedQuickSort	Найкращий	112	250.01K	1.18
MultiThreadedMergeSort	Найгірший	800080	250.0K	3.45
MultiThreadedMergeSort	Рівномірний	800080	300.00K	5.13
MultiThreadedMergeSort	Нормальний	800080	300.00K	5.46
MultiThreadedMergeSort	Найкращий	800080	250.0K	3.88
MultiThreadedHeapSort	Найгірший	0	8.07M	7.54
MultiThreadedHeapSort	Рівномірний	0	7.74M	8.44
MultiThreadedHeapSort	Нормальний	0	7.74M	8.43
MultiThreadedHeapSort	Найкращий	0	7.42M	6.72

double, $N = 200.0K$

Алгоритм	Умова	V (Б)	K	T (мс)
QuickSort	Найгірший	0	18.16M	3.37
QuickSort	Рівномірний	0	10.39M	10.97
QuickSort	Нормальний	0	10.38M	11.49
QuickSort	Найкращий	0	9.04M	1.39
MergeSort	Найгірший	28302848	8.88M	13.61
MergeSort	Рівномірний	28302848	10.35M	25.87
MergeSort	Нормальний	28302848	10.35M	25.37
MergeSort	Найкращий	28302848	8.81M	13.74
HeapSort	Найгірший	0	17.12M	18.30
HeapSort	Рівномірний	0	16.49M	19.95
HeapSort	Нормальний	0	16.49M	20.38
HeapSort	Найкращий	0	15.84M	16.54
ShellSort	Найгірший	0	12.98M	2.48
ShellSort	Рівномірний	0	23.33M	19.46
ShellSort	Нормальний	0	23.31M	24.31
ShellSort	Найкращий	0	9.60M	1.95
MultiThreadedQuickSort	Найгірший	112	500.01K	6.12
MultiThreadedQuickSort	Рівномірний	112	511.36K	5.59
MultiThreadedQuickSort	Нормальний	112	474.76K	4.99
MultiThreadedQuickSort	Найкращий	112	500.01K	1.98
MultiThreadedMergeSort	Найгірший	1600080	500.0K	8.47
MultiThreadedMergeSort	Рівномірний	1600080	600.00K	10.73
MultiThreadedMergeSort	Нормальний	1600080	600.00K	10.18
MultiThreadedMergeSort	Найкращий	1600080	500.0K	6.59
MultiThreadedHeapSort	Найгірший	0	17.12M	15.39
MultiThreadedHeapSort	Рівномірний	0	16.49M	19.00
MultiThreadedHeapSort	Нормальний	0	16.49M	925.38
MultiThreadedHeapSort	Найкращий	0	15.84M	49.12

double, $N = 500.0K$

Алгоритм	Умова	V (Б)	K	T (мс)
QuickSort	Найгірший	0	49.91M	10.08
QuickSort	Рівномірний	0	27.92M	30.86
QuickSort	Нормальний	0	28.03M	32.14
QuickSort	Найкращий	0	23.67M	4.15
MergeSort	Найгірший	75805696	23.73M	32.92
MergeSort	Рівномірний	75805696	27.79M	71.58
MergeSort	Нормальний	75805696	27.79M	19096
HeapSort	Найгірший	0	45.90M	47.77
HeapSort	Рівномірний	0	44.47M	55.86

(продовження)

Алгоритм	Умова	V (Б)	К	T (мс)
HeapSort	Нормальний	0	44.47M	54.87
HeapSort	Найкращий	0	42.98M	43.21
ShellSort	Найгірший	0	33.86M	8.56
ShellSort	Рівномірний	0	66.38M	67.37
ShellSort	Нормальний	0	66.26M	60.82
ShellSort	Найкращий	0	25.50M	4.76
MultiThreadedQuickSort	Найгірший	112	1.25M	9.52
MultiThreadedQuickSort	Рівномірний	112	1.28M	10.60
MultiThreadedQuickSort	Нормальний	112	1.21M	9.73
MultiThreadedQuickSort	Найкращий	112	1.25M	3.96
MultiThreadedMergeSort	Найгірший	4000080	1.25M	17.29
MultiThreadedMergeSort	Рівномірний	4000080	1.50M	25.79
MultiThreadedMergeSort	Нормальний	4000080	1.50M	23.10
MultiThreadedMergeSort	Найкращий	4000080	1.25M	15.21
MultiThreadedHeapSort	Найгірший	0	45.90M	54.83
MultiThreadedHeapSort	Рівномірний	0	44.47M	67.38
MultiThreadedHeapSort	Нормальний	0	44.47M	51.99
MultiThreadedHeapSort	Найкращий	0	42.98M	41.11

double, $N = 1.0M$

Алгоритм	Умова	V (Б)	К	T (мс)
QuickSort	Найгірший	0	106.44M	23.45
QuickSort	Рівномірний	0	58.78M	66.09
QuickSort	Нормальний	0	58.77M	63.13
QuickSort	Найкращий	0	49.85M	7.28
HeapSort	Найгірший	0	97.06M	91.62
HeapSort	Рівномірний	0	93.94M	119.19
HeapSort	Нормальний	0	93.94M	120.24
HeapSort	Найкращий	0	91.00M	93.24
ShellSort	Найгірший	0	71.72M	16.89
ShellSort	Рівномірний	0	153.39M	128.46
ShellSort	Нормальний	0	153.29M	128.81
ShellSort	Найкращий	0	54.00M	10.13
MultiThreadedQuickSort	Найгірший	112	2.50M	19.89
MultiThreadedQuickSort	Рівномірний	112	2.40M	21.11
MultiThreadedQuickSort	Нормальний	112	2.52M	20.45
MultiThreadedQuickSort	Найкращий	112	2.50M	8.34
MultiThreadedMergeSort	Найгірший	8000080	2.50M	30.00
MultiThreadedMergeSort	Рівномірний	8000080	3.00M	49.74
MultiThreadedMergeSort	Нормальний	8000080	3.00M	46.46
MultiThreadedMergeSort	Найкращий	8000080	2.50M	28.03
MultiThreadedHeapSort	Найгірший	0	97.06M	89.17
MultiThreadedHeapSort	Рівномірний	0	93.94M	115.59
MultiThreadedHeapSort	Нормальний	0	93.94M	114.05

(продовження)

Алгоритм	Умова	V (Б)	К	T (мс)
MultiThreadedHeapSort	Найкращий	0	91.00M	89.72

7 Висновок

- **Узгодження з теорією.** InsertionSort: у найкращому випадку $\text{BigO} \approx 0.38N$ (відповідає $O(n)$), у найгіршому/рівномірному/нормальному $\approx 0.07\text{--}0.13 N^2$ ($O(n^2)$). SelectionSort: $\approx 0.61\text{--}0.67 N^2$ у всіх умовах — відповідає $O(n^2)$. HeapSort: BigO $3.2\text{--}3.8 N \lg N$ — узгоджується з $O(n \log n)$. ShellSort: $0.77\text{--}4.48 N \lg N$. QuickSort у найгіршому, рівномірному та найкращому: $0.89 N \lg N$, $2.77 N \lg N$, $0.39 N \lg N$. CountingSort і RadixSort: лінійні або близькі до $N \lg N$. MultiThreadedHeapSort і MultiThreadedBubbleSort (Best): BigO узгоджується з теорією.
- **Невідповідності теорії (артефакти апроксимації).** MergeSort — бенчмарк підібрав модель лінійну за N ($50.15N$, $73.32N$ тощо) замість $O(n \log n)$; *не відповідає теорії* (обмежений діапазон N). QuickSort (Normal) — BigO $45.40N$ замість $O(n \log n)$; *помилковий підбор моделі*. BubbleSort (Uniform, Normal) — BigO $0.00N^3$; теорія $O(n^2)$ — *артефакт*. MultiThreadedMergeSort — дуже великі коефіцієнти при $\lg N$ ($85328 \lg N$ тощо); *артефакт апроксимації*. MultiThreadedQuickSort — `real_time` як $12.55N$; теорія $O(n \log n)$; *у діапазоні 10K–100K може виглядати лінійно через паралелізм*.
- Найшвидшими на великих об'ємах залишаються CountingSort, RadixSort (для `int`), QuickSort, MergeSort, HeapSort та їхні багатопотокові варіанти; алгоритми $O(n^2)$ та їхні паралельні версії на великих N значно повільніші, з таймаутами при $N \geq 10^5\text{--}2 \cdot 10^5$.
- Багатопотокові QuickSort, MergeSort, HeapSort дають помітне прискорення за `real_time`; паралельні InsertionSort і BubbleSort часто гірші за однопотокові.
- Додаткова пам'ять V узгоджується з оцінками: MergeSort/MultiThreadedMergeSort — значне V , RadixSort/CountingSort — помірне, QuickSort/HeapSort — незначне або нульове.
- Для типів `int` і `double` результати подібні; графіки часу від N підтверджують очікувані залежності за складністю.

8 Код програми та скрінш

Реалізація, бенчмарки та додаток для графіків доступні в репозиторії:

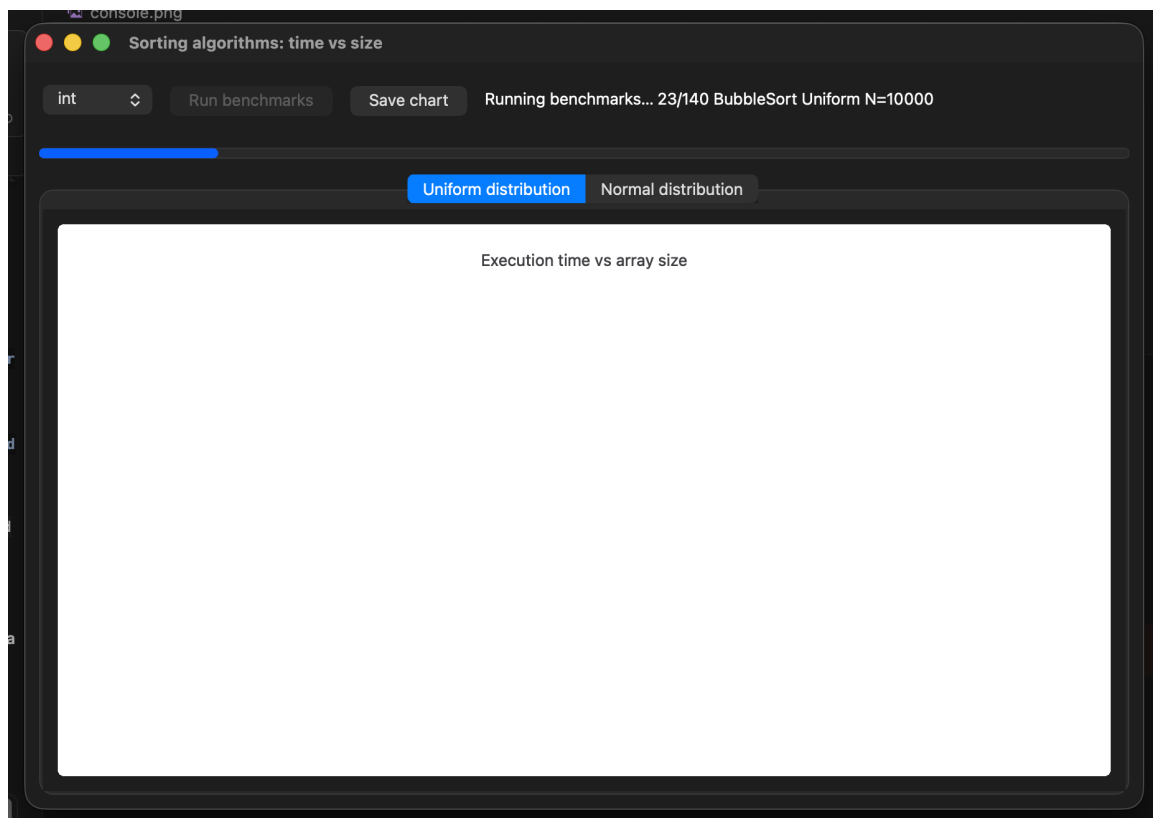
<https://github.com/kshchuk/Sorting-Algorithms>

Скріншоти консолі та GUI

Консольний вивід бенчмарку (стратегії, прогрес, таблиці CSV):

```
[int] 119/392 BubbleSort N=50000 Uniform ... done
[int] 120/392 BubbleSort N=50000 Normal ... done
[int] 121/392 BubbleSort N=50000 Best ... done
[int] 122/392 BubbleSort N=100000 Worst ... done
[int] 123/392 BubbleSort N=100000 Uniform ...
done
[int] BubbleSort timeout, skipping N>=200000
[int] 136/392 SelectionSort N=10000 Worst ... done
[int] 137/392 SelectionSort N=10000 Uniform ... done
[int] 138/392 SelectionSort N=10000 Normal ... done
[int] 139/392 SelectionSort N=10000 Best ... done
[int] 140/392 SelectionSort N=20000 Worst ... done
[int] 141/392 SelectionSort N=20000 Uniform ... done
[int] 142/392 SelectionSort N=20000 Normal ... done
[int] 143/392 SelectionSort N=20000 Best ... done
[int] 144/392 SelectionSort N=50000 Worst ... done
[int] 145/392 SelectionSort N=50000 Uniform ... done
[int] SelectionSort timeout, skipping N>=100000
[int] 162/392 ShellSort N=10000 Worst ... done
[int] 163/392 ShellSort N=10000 Uniform ... done
[int] 164/392 ShellSort N=10000 Normal ... done
[int] 165/392 ShellSort N=10000 Best ... done
[int] 166/392 ShellSort N=20000 Worst ... done
[int] 167/392 ShellSort N=20000 Uniform ... done
```

Графічний додаток `sorting_charts` (вибір типу даних, запуск бенчмарків, графіки часу від N):



Графіки залежності часу від розміру масиву

Графіки побудовано за результатами експериментів для розподілів Uniform та Normal (типи `int` та `double`).

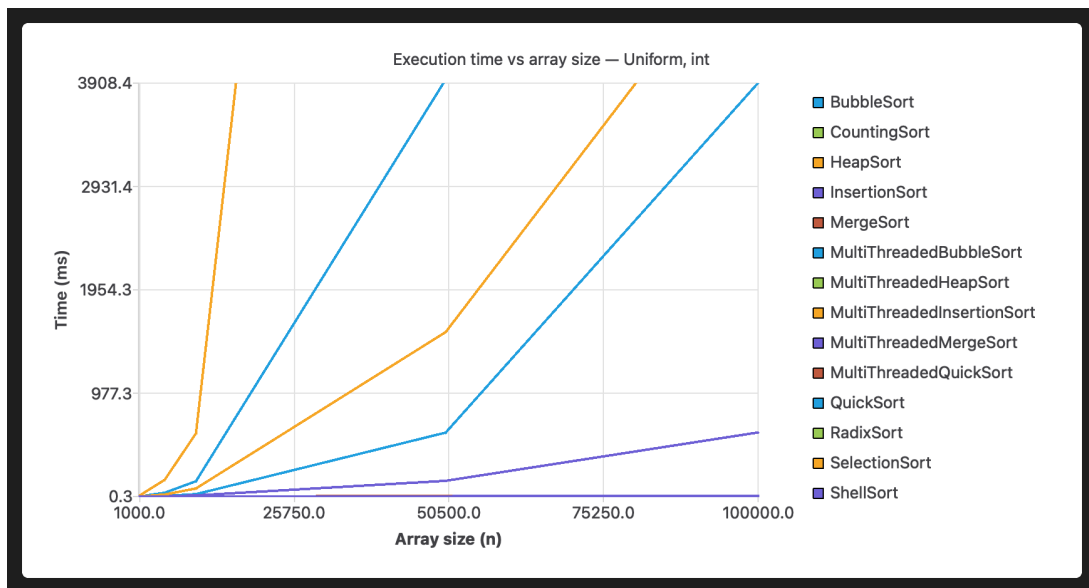


Рис. 1: Рівномірний розподіл, тип `int`.

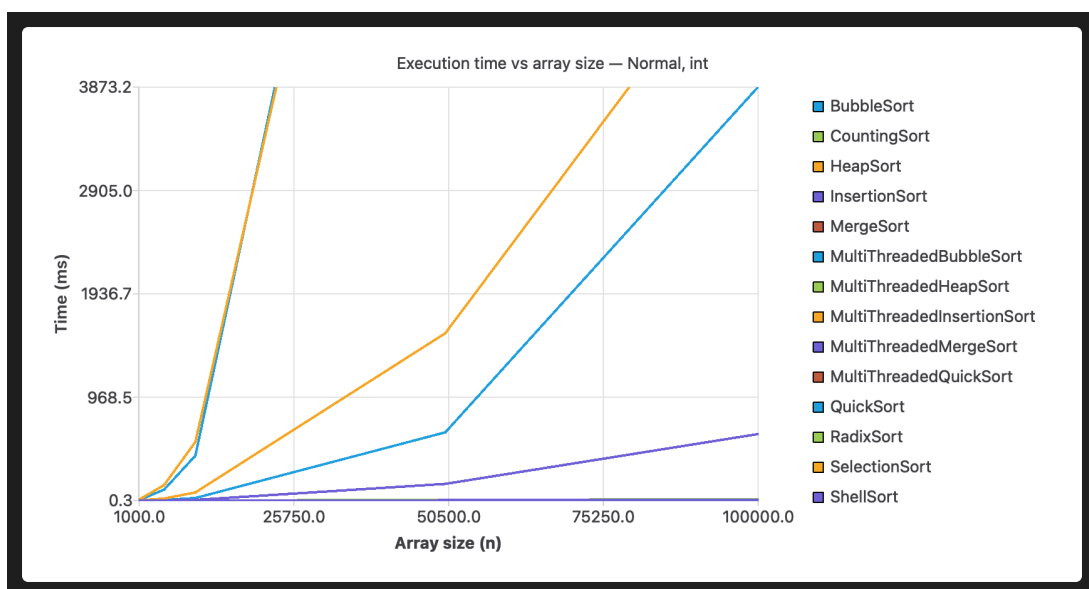


Рис. 2: Нормальний розподіл, тип `int`.

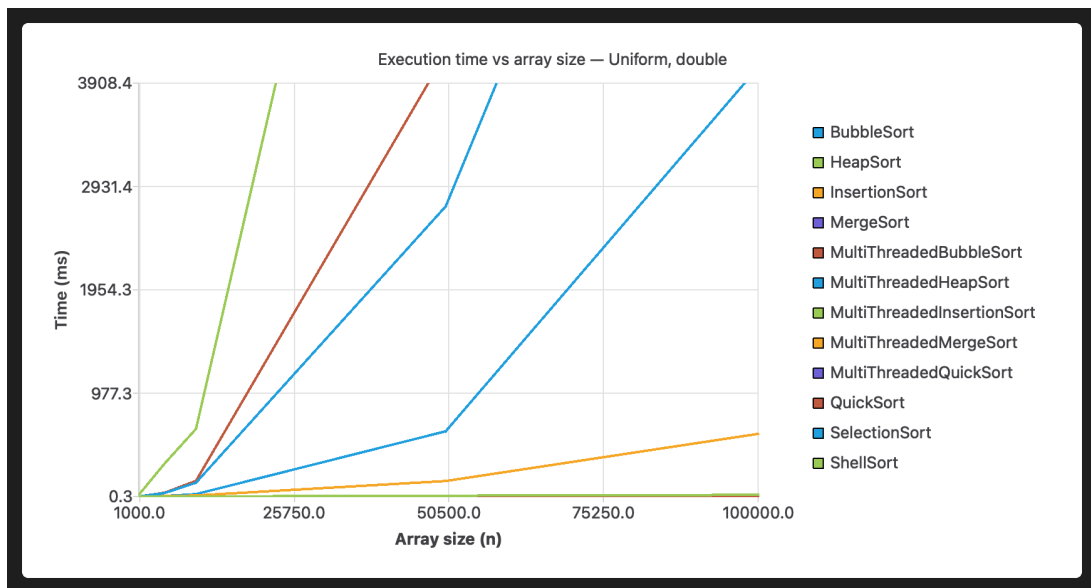


Рис. 3: Рівномірний розподіл, тип double.

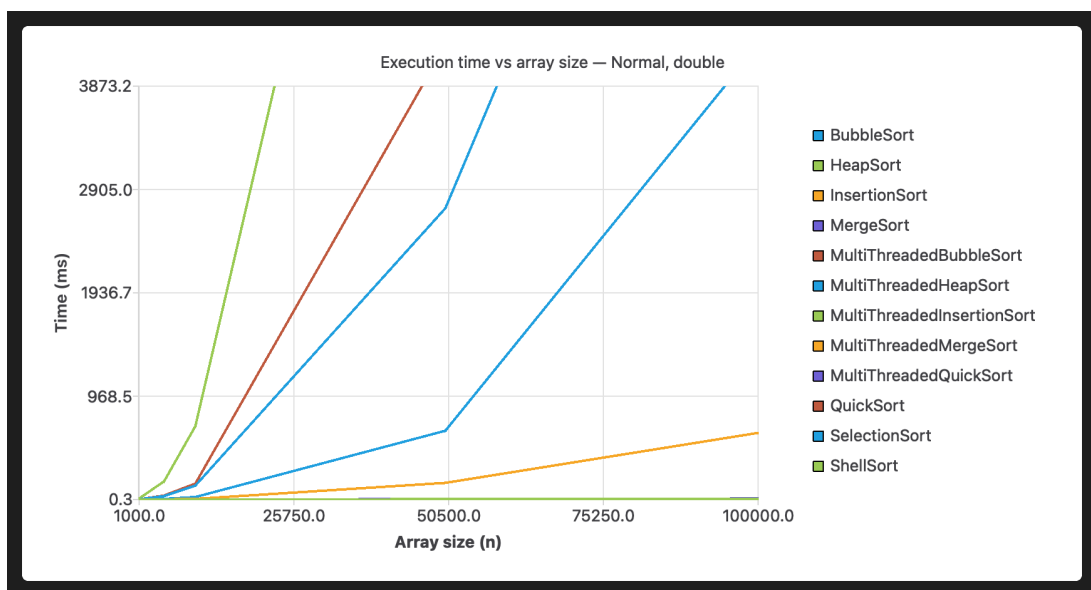


Рис. 4: Рівномірний та нормальний розподіли (порівняння).