

Лабораторна робота № 2

МЕТОДИ СОРТУВАННЯ МАСИВІВ

Мета роботи: познайомитися з роботою поширених методів сортування, з критеріями та методикою їх порівняння.

2.1 Теоретичні відомості

ПРОСТІ МЕТОДИ СОРТУВАННЯ

Розглянемо декілька методів впорядкування елементів масиву, які широко використовуються у практичному програмуванні.

Сортування вставками

Даний метод сортування називається сортування вставками, так як на i -му етапі відбувається "вставка" i -ого елемента $a[i]$ в потрібну позицію серед елементів $a[1]$, $a[2]$, ..., $a[i-1]$, які вже впорядковані. Після цієї вставки перші i елементів будуть впорядковані.

Масив до впорядкування	22	20	-1	-40	88	-75	-22
Перший перегляд масиву	20	22	-1	-40	88	-75	-22
Другий перегляд масиву	-1	20	22	-40	88	-75	-22
Третій перегляд масиву	-40	-1	20	22	88	-75	-22
Четвертий перегляд масиву	-40	-1	20	22	88	-75	-22
П'ятий перегляд масиву	-75	-40	-1	20	22	88	-22
Шостий перегляд масиву	-75	-40	-22	-1	20	22	88

Рис 3. Сортування вставками

Серед критеріїв, за якими порівнюються методи сортування масивів, розрізняють:

- **C** (від **compare**) - кількість порівнянь ключів між собою;
- **M** (від **move**) - кількість операцій перезапису елементів з місця на місце у оперативній пам'яті або кількість обмінів;
- **T** (від **time**) - загальний час роботи процедури.

Аналіз. Кількість перевірок на i -му кроці дорівнює щонайбільше $i-1$, щонайменше 1 , тому в середньому - $i/2$. Тому в середньому загальна кількість перевірок

$$C_{\text{сер}} = \sum_{i=2}^n \frac{i}{2} = \frac{(2+3+\dots+n)}{2} = \frac{(2+n)(n-1)}{4} = O(n^2)$$

При цьому

$$C_{\min} = n - 1; \quad C_{\max} = \frac{(n-1)n}{2}.$$

Кількість обмінів **M** дорівнює щонайбільше **i**, щонайменше **0** на **i**-му кроці, тобто **i/2** у середньому. Тому

$$M_{\min} = 0; \quad M_{\max} = \frac{(n-1)n}{2}; \quad M_{\text{ср}} = \frac{(n-1)n}{4}.$$

Метод бульбашкового сортування

Метод "бульбашкового сортування" ґрунтується на перестановці сусідніх елементів. Для впорядкування елементів масиву здійснюються повторні проходи по масиву. Переміщення елементів масиву здійснюється таким чином: масив переглядається зліва направо, порівнюються два сусідніх елементи; якщо елементи в парі розміщені в порядку зростання, вони лишаються без змін, а якщо ні - міняються місцями.

В результаті першого проходу найбільше число буде поставлене в кінець масиву. У другому проході такі операції виконуються над елементами з першого до (N-1)-ого, у третьому - від першого до (N-2)-ого і т.д. Впорядкування масиву буде закінчено, якщо при проході масиву не виконається жодної перестановки елементів масиву. Факт перестановки фіксується за допомогою деякої змінної, яка на початку має значення 0 і набуває значення 1 тоді, коли виконається перестановка в якій-небудь парі.

Масив до впорядкування	22	20	-1	-40	88	-75	-22
Перший перегляд масиву	20	-1	-40	22	-75	-22	88
Другий перегляд масиву	-1	-40	20	-75	-22	22	88
Третій перегляд масиву	-40	-1	-75	-22	20	22	88
Четвертий перегляд масиву	-40	-75	-22	-1	20	22	88
П'ятий перегляд масиву	-75	-40	-22	-1	20	22	88

Рис 1. Бульбашкове сортування

Кількість порівнянь для "бульбашкового методу":

$$C = \frac{(n-1)n}{2};$$

обмінів:

$$M_{\min} = 0; \quad M_{\max} = \frac{(n-1)n}{2}; \quad M_{\text{ср}} = \frac{(n-1)n}{4}.$$

Сортування методом вибору

Даний метод сортування передбачає наступні дії: масив переглядається перший раз, знаходиться мінімальний елемент цього масиву, який міняється місцями з першим елементом. Другий раз масив переглядається, починаючи з другого елементу. Знову знаходиться мінімальний елемент, який міняється місцями з другим елементом масиву.

Даний процес виконується до тих пір, поки не буде поставлено на місце N-1 елемент.

Масив до впорядкування	22	20	-1	-40	88	-75	-22
Перший перегляд масиву	-75	20	-1	-40	88	22	-22
Другий перегляд масиву	-75	-40	-1	20	88	22	-22
Третій перегляд масиву	-75	-40	-22	20	88	22	-1
Четвертий перегляд масиву	-75	-40	-22	-1	88	22	20
П'ятий перегляд масиву	-75	-40	-22	-1	20	22	88
Шостий перегляд масиву	-75	-40	-22	-1	20	22	88

Рис 2. Сортуння методом вибору

Оцінимо якісні характеристики програми. На **i**-му кроці робиться **(n-i)** перевірок з елементами **i+1, i+2, ..., n**.

Тому

$$C = (n-1) + (n-2) + \dots + 2 + 1 = \frac{(n-1)n}{2}.$$

$$M_{сер} = \frac{0 + (n-1)}{2} = \frac{n-1}{2}$$

МОДИФІКОВАНІ МЕТОДИ СОРТУВАННЯ

Головний недолік простих методів - обмін ведеться в основному між сусідніми елементами. Тому бажано робити якомога ширші обміни.

Метод Шелла

Метод Шелла (Shell D.L., 1959) - метод сортуння вставками (включеннями) з відстанями, що зменшуються.

Візьмемо для прикладу масив:

60	16	41	06	59	79	34	15
----	----	----	----	----	----	----	----

На першому етапі масив уявно ділиться на підмасиви з елементами, що, скажімо, відстоять один від одного на 4 елементи:

1.

60				59			
----	--	--	--	----	--	--	--

2.

	16				79		
--	----	--	--	--	----	--	--

3.

		41				34	
--	--	----	--	--	--	----	--

4.

			06				15
--	--	--	----	--	--	--	----

Кожен з них впорядковується окремо:

1.

59				60			
----	--	--	--	----	--	--	--

2.

	16				79		
--	----	--	--	--	----	--	--

3.

		34				41	
--	--	----	--	--	--	----	--

4.

			06				15
--	--	--	----	--	--	--	----

Сортування робиться у кожному підмасиві вставками (методом простого включення).

На другому етапі підмасиви утворюються елементами через один:

1.

59		34		60		41	
----	--	----	--	----	--	----	--

2.

	16		06		79		15
--	----	--	----	--	----	--	----

Одержуємо

1.

34		41		59		60	
----	--	----	--	----	--	----	--

2.

	06		15		16		79
--	----	--	----	--	----	--	----

Після цього весь масив сортується разом. За рахунок попередніх етапів він виявляється вже близьким до відсортованого, тому обмінів необхідно вже не так багато.

Аналіз та експериментальні дослідження методу показують, що цей метод дає кращі результати, якщо розподіл на підмасиви роблять кроками, що не є степенями двійки, а, навпаки, не є множниками один одного.

Рекомендованими є такі послідовності кроків:

$$\begin{aligned} h_{k-1} &= 3h_k + 1, \quad h_x = 1, \quad K = [\log_3 n] - 1 & \text{або} \\ h_{k-1} &= 2h_k + 1, \quad h_x = 1, \quad K = [\log_2 n] - 1. \end{aligned}$$

У останньому випадку кількість необхідних операцій пропорційна $n^{6/5} = n^{1.2}$.

ШВИДКЕ СОРТУВАННЯ

В загальному алгоритм швидкого сортування можна описати так:
`quickSort`

- Вибрати опорний елемент p
- Розділити масив по цьому елементу (реорганізувати масив таким чином, щоб всі елементи, менші або рівні опорному, виявилися зліва від нього, а всі елементи, більші опорного, – справа від нього)
- Якщо підмасив зліва від p містить більше одного елемента, викликати `quickSort` для нього (тобто повторити рекурсивно для підмасиву зліва від p)
- Якщо підмасив справа від p містить більше одного елемента, викликати `quickSort` для нього (тобто повторити рекурсивно для підмасиву справа від p)

Часто в якості опорного елемента пропонується вибрати медіану (середину масиву). Однак можна підібрати приклад, при якому алгоритм з вибором медіани в якості опорного елемента буде видавати неправильну відповідь. Відомі стратегії: вибирати постійно один і той самий елемент, наприклад, перший або останній; вибрати елемент випадковим чином.

Недолік вибору в якості опорного одного із крайніх елементів масиву — при передачі параметром уже відсортованого масиву такий вибір призводить до найгіршого випадку.

Недолік вибору опорного елемента випадковим чином — залежність швидкості алгоритму від реалізації генератора псевдовипадкових чисел. Якщо генератор працює повільно і видає погані послідовності псевдовипадкових чисел, можлива затримка роботи алгоритму.

Оцінювання середньостатистичних значень M та C є нелегкою задачею з огляду на необхідність використання апарату теорії ймовірностей, але обидві величини будуть порядку

$$\sim N \log_2 N.$$

2.2. Порядок виконання роботи

1. Написати програму, що реалізує один з простих методів сортування (згідно з номером варіанту).
2. Написати програму, що реалізує метод Шелла або швидкого сортування (згідно з номером варіанту).

3. Згенерувати три масиви з випадковими елементами типу **Integer** довжиною 100, 1000 та 10000 елементів, відповідно.
4. Відсортувати одержані масиви за збільшенням елементів, визначивши при цьому такі параметри:
 - кількість порівнянь;
 - кількість обмінів;
 - фактичний час роботи,

Швидкодія сучасних комп'ютерів може привести до того, що час роботи процедури буде дорівнювати нулеві з точністю, яку забезпечує системний таймер. Тоді варто запустити її багато разів у циклі для різних масивів, а потім усереднити результат.

5. Оформити звіт, навівши одержані експериментальні дані та теоретичні оцінки у вигляді таблиць.

Звіт має містити:

- a. текст програми, яку Ви написали;
- b. надрукований масив на 100 елементів до сортування та після нього;
- c. теоретичні оцінки для кількостей операцій для методів Вашого варіанту;
- d. результати експериментального дослідження та - обов'язково - їхній аналіз.

Останні два пункти можна викласти у таблиці:

Результати порівняння методів сортування										
N	Назва методу					Назва методу				
	К-ть копіювань (М)		К-ть порівнянь (С)		Час (Т)	К-ть копіювань (М)		К-ть порівнянь (С)		Час (Т)
	Теорет.	Експерим.	Теорет.	Експерим.		Теорет.	Експерим.	Теорет.	Експерим.	
100										
1000										
10000										

Для методу Шелла відома лише порядкова оцінка. Це означає, що експериментальні величини можуть відрізнятися від теоретичних у багато разів, але відношення Мексп / Мтеор та Сексп / Стеор мають бути приблизно постійними незалежно від значення N. Порахуйте ці відношення, занесіть до таблиці та проаналізуйте.

2.3. Варіанти завдань

№ варіанту	Метод	
	"простий"	"складний"
1	"Вставками"	метод Шелла
2	"Вибору"	метод Шелла
3	"Бульбашки"	метод Шелла
4	"Вставками"	метод швидкого сортування
5	"Вибору"	метод швидкого сортування
6	"Бульбашки"	метод швидкого сортування