

Практична робота № 3

Тема: Реалізація алгоритмів сортування

Мета роботи: закріплення навичок роботи з масивами, освоєння найпростіших алгоритмів сортування.

Теоретичні відомості

Метод бульбашки для масиву $a[1], a[2], \dots, a[n]$ працює так. Починаючи з кінця масиву, порівнюємо два сусідні елементи ($a[n]$ і $a[n-1]$). Якщо виконується умова $a[n-1] > a[n]$, то значення елементів міняються місцями. Процес триває для $a[n-1]$ і $a[n-2]$ і т. д., поки не буде виконано порівняння $a[2]$ і $a[1]$. Зрозуміло, що після цього на місці $a[1]$ виявиться елемент масиву з найменшим значенням. На другому кроці процес повторюється, але останніми порівнюються $a[3]$ і $a[2]$. На останньому кроці будуть порівнюватися тільки значення $a[n]$ і $a[n-1]$. Зрозуміла аналогія з бульбашкою, оскільки найменші елементи (самі «легкі») поступово «спливають» до верхньої межі масиву.

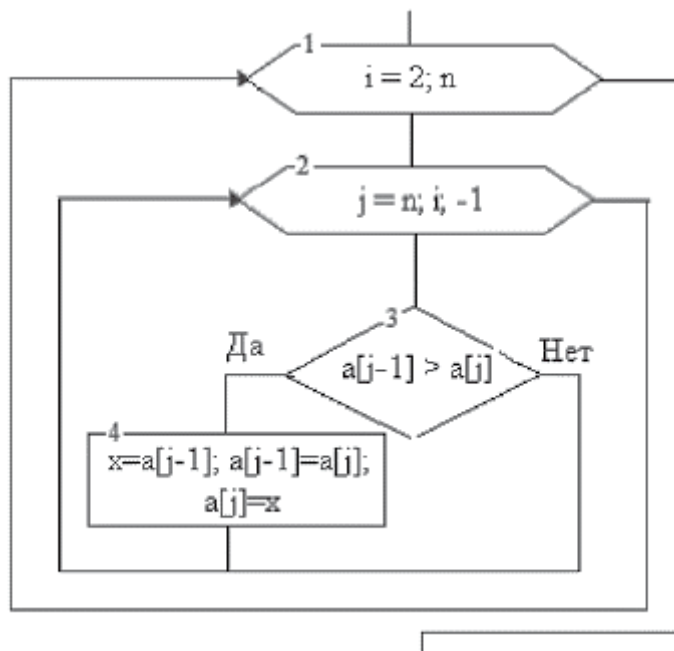


Рис.1 – Алгоритм сортування методом бульбашки

Блок 1. Арифметичний цикл. Значення параметра циклу i визначає кількість елементів для порівняння у внутрішньому циклі.

Блок 2. Завдання номерів елементів для порівняння.

Блок 3. Порівняння двох сусідніх елементів.

Блок 4. Виконується обмін елементів масиву при виході «Так» блоку 3.

Розглянемо сортування масиву 32 64 9 30 87 14 2 76. Виконання алгоритму для $i = 2$ наведено на рис. 2.

i	j	$a[j - 1] > a[j]$?	$a[j - 1], a[j]$	Масив
2	8	$2 > 76$? «Нет»		32 64 9 30 87 14 2 76
	7	$14 > 2$? «Да»	$a[6] = 2, a[7] = 14$	32 64 9 30 87 2 14 76
	6	$87 > 2$? «Да»	$a[5] = 2, a[6] = 87$	32 64 9 30 2 87 14 76
	5	$30 > 2$? «Да»	$a[4] = 2, a[5] = 30$	32 64 9 2 30 87 14 76
	4	$9 > 2$? «Да»	$a[3] = 9, a[4] = 2$	32 64 2 9 30 87 14 76
	3	$64 > 2$? «Да»	$a[2] = 2, a[3] = 64$	32 2 64 9 30 87 14 76
	2	$32 > 2$? «Да»	$a[1] = 2, a[2] = 32$	2 32 64 9 30 87 14 76

Рис. 2 – Приклад сортування масиву методом бульбашкою

Після цього кроку значення 2 встало на своє місце, і алгоритм повторюється для $i = 3, 4, \dots, n$.

Практичне завдання

Написати програму, в якій:

1. Згенерувати одновимірний масив цілих чисел розмірністю згідно варіанту.
2. Елементи масиву задати випадковим чином в діапазоні 0 ... 1000.
3. Запам'ятати цей масив. Виконати друк цього масиву на екран.
4. Виконати обробку масиву відповідно до варіанту. При написанні програми використати метод сортування бульбашкою.
5. Під час сортування бульбашкою визначити кількість операцій.
6. Виконати друк перетвореного масиву на екран.
7. Здійснити пошук вказаного (з клавіатури) елемента у масиві, використовуючи лінійний пошук).
8. Здійснити пошук вказаного (з клавіатури) елемента у масиві, використовуючи бінарний пошук).

Оформити звіт.

Контрольні питання

1. Для чого потрібні алгоритми сортування?
2. З яких основних частин складається будь-який алгоритм сортування?
3. Які основні параметри алгоритмів сортування ви знаєте?
4. Поясніть принцип роботи сортування вибором.
5. Поясніть принцип роботи сортування методом простого обміну

Таблиця 1 – Варіанти індивідуальних завдань

№	Завдання
1	Є одномірний масив довжиною $N = 33$. Упорядкувати масив таким чином, щоб елементи, що знаходяться на парних позиціях розташовувалися за спаданням, а на непарних позиціях - по зростанню.
2	Відсортувати одновимірний масив довжиною $N = 67$ за зростанням.
3	Є одномірний масив довжиною $N = 45$. Відсортувати за спаданням елементи масиву, які є парними числами.
4	Є одномірний масив довжиною $N = 38$. Упорядкувати масив таким чином, щоб елементи, що знаходяться на парних позиціях розташовувалися по зростанню, а на непарних позиціях - за спаданням.
5	Є одномірний масив довжиною $N = 32$. Відсортувати за зростанням ті елементи масиву, які розташовуються на непарних позиціях.