

# **Лекція 12. Сортування масивів.**

---

**Сортування масиву** – це впорядкування елементів масиву, щоб вони розташовувались в порядку зростання чи спадання.

*масив*

<b>-5</b>	<b>6</b>	<b>3</b>	<b>-7</b>	<b>3</b>	<b>1</b>	<b>10</b>	<b>1</b>	<b>-3</b>	<b>5</b>	<b>8</b>
-----------	----------	----------	-----------	----------	----------	-----------	----------	-----------	----------	----------

*відсортований масив*

<b>-7</b>	<b>-5</b>	<b>-3</b>	<b>1</b>	<b>1</b>	<b>3</b>	<b>3</b>	<b>5</b>	<b>6</b>	<b>8</b>	<b>10</b>
-----------	-----------	-----------	----------	----------	----------	----------	----------	----------	----------	-----------

---

# масив

<b>-5</b>	<b>6</b>	<b>3</b>	<b>-7</b>	<b>3</b>	<b>1</b>	<b>10</b>	<b>1</b>	<b>-3</b>	<b>5</b>	<b>8</b>
-----------	----------	----------	-----------	----------	----------	-----------	----------	-----------	----------	----------

*відсортований масив за зростанням*

<b>-7</b>	<b>-5</b>	<b>-3</b>	<b>1</b>	<b>1</b>	<b>3</b>	<b>3</b>	<b>5</b>	<b>6</b>	<b>8</b>	<b>10</b>
-----------	-----------	-----------	----------	----------	----------	----------	----------	----------	----------	-----------

*відсортований масив за спаданням*

<b>10</b>	<b>8</b>	<b>6</b>	<b>5</b>	<b>3</b>	<b>3</b>	<b>1</b>	<b>1</b>	<b>-3</b>	<b>-5</b>	<b>-7</b>
-----------	----------	----------	----------	----------	----------	----------	----------	-----------	-----------	-----------

---

# Є різні методи сортування масивів.

Найвідоміші:

- 1) Бульбашкове сортування
  - 2) Сортування вибором
  - 3) Сортування вставками
  - 4) Сортування Шелла
  - 5) Швидке сортування
  - 6) Пірамідальне сортування
  - 7) Сортування гребінцем
  - 8) інші алгоритми
-

# Бульбашкове сортування

Ідея алгоритму:

- у циклі виконується прохід зліва направо по елементам масиву і на кожній кроці порівнюються два сусідні елементи: **поточний та наступний за ним**;
  - якщо ці два елементи стоять в неправильному порядку, то вони міняються місцями;
  - якщо хоча б на одній ітерації циклу відбувся обмін двох елементів, то даний процес повторюється ще раз спочатку.
-

# Бульбашкове сортування


Масив до впорядку вання	22	20	-1	-40	88	-75	-22
1	20	-1	-40	22	-75	-22	88
2	-1	-40	20	-75	-22	22	88
3	-40	-1	-75	-22	20	22	88
4	-40	-75	-22	-1	20	22	88
5	-75	-40	-22	-1	20	22	88

# Бульбашкове сортування

```
int n = 7; int tmp, fl;
int array[] = {22, 20, -1, -40, 88, -75, -22};

do
{
    fl = 0;
    for (int i = 1; i < n; i++)
        if (array[i - 1] > array[i]){
            tmp = array[i]; array[i] = array[i-1];
            array[i-1] = tmp;
            fl = 1;
        }
} while (fl);
```

---



$\{ 2, -2, -1, -4, 8, -7, 5, -2, 2 \}$

---



# *Бульбашкове сортування*

*Повний код програми розміщено за адресою:*

<https://1drv.ms/w/s!AvLKc6r1gw0VtXIk8k2ukY5srHsT>

*Відео візуалізація бульбашкового сортування:*

<https://www.youtube.com/watch?v=lyZQPjUT5B4>

---

# Сортування вставками

Ідея алгоритму:

виконується  $n-1$  кроків, на кожному з яких здійснюється вставка  $i$ -ого елемента  $arr[i]$  ( $i = 1, \dots, n-1$ ) в потрібну позицію серед елементів  $arr[0], \dots, arr[i-1]$ , які вже є впорядкованими

---


# Сортування вставками

Масив до сортування	22	20	-1	-40	88	-75	-22
1	20	22					
2	-1	20	22				
3	-40	-1	20	22			
4	-40	-1	20	22	88		
5	-75	-40	-1	20	22	88	
6	-75	-40	-22	-1	20	22	88

# Сортування вставками


```
int n = 7; int tmp, fl;  
int array[] = { 22, 20, -1, -40, 88, -75, -22 };  
  
for (int i = 1; i < n; i++)  
{  
    tmp = array[i];  
    for (int j=i-1; j>=0 && array[j]>tmp; j--)  
    {  
        array[j + 1] = array[j];  
        array[j] = tmp;  
    }  
}
```

---



$\{ 2, -2, -1, -4, 8, -7, 5, -2, 2 \}$

---



*Повний код програми розміщено  
за адресою:*

<https://1drv.ms/w/s!AvLKc6r1gw0VtX1W4oYV-chU9MzF>

---

# Сортування методом Шелла

Ідея методу Шелла полягає в порівнянні елементів, що стоять не тільки поруч, але і на певній відстані один від одного.

- На першому етапі масив уявно ділиться на підмасиви наприклад  $size = 8$ ,  
 $step = size / 2 = 4$ ,  
кожен з них впорядковується окремо.
  - На другому етапі підмасиви утворюються елементами через один.
-

# Сортування методом Шелла

<b>Масив до сортування</b>	<b>22</b>	<b>20</b>	<b>-1</b>	<b>-40</b>	<b>88</b>	<b>-75</b>	<b>-22</b>
<b>1</b>	-40	20	-75	-22	88	-1	22
<b>2</b>	-75	-40	-22	-1	20	22	88




# Сортування методом Шелла

```
int size = 7, tmp;
int array[] = {22, 20, -1, -40, 88, -75, -22 };

int step = size / 2;
while (step > 0){
    for (int i = 0; i < (size - step); i++){
        int j = i;
        while (j >= 0 && array[j] > array[j + step]){
            tmp = array[j];
            array[j] = array[j + step];
            array[j + step] = tmp;
            j--;
        }
    }
    step = step / 2;
}
```

---



$\{ 2, -2, -1, -4, 8, -7, 5, -2, 2 \}$

---

# Метод швидкого сортування (QuickSort)

Алгоритм швидкого сортування можна описати так:

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

# Метод швидкого сортування (QuickSort)

Масив до сортування	22	20	-1	-40	88	-75	-22
	-75	-40	-1	20	88	22	-22
	-75	-40	-1	20	-22	22	88
	-75	-40	-1	-22	20	22	88
	-75	-40	-22	-1	20	22	88

# Метод швидкого сортування (QuickSort)

Масив до сортування	22	20	-1	-40	88	-75	-22
	20	-1	-40	-75	-22	22	88
	-75	-40	-1	20	-22	22	88
	-75	-40	-1	-22	20	22	88

```
void qs(int *arr, int first, int last){
    if (first < last){
        int left = first, right = last, middle = arr[(left + right)/2];
        do{
            while (arr[left] < middle) left++;
            while (arr[right] > middle) right--;
            if (left <= right){
                int tmp = arr[left]; arr[left] = arr[right];
                arr[right] = tmp;
                left++; right--;
            }
        } while (left <= right);
        qs(arr, first, right); qs(arr, left, last);
    }
}

int main() {
    int n = 7; int array[] = {22, 20, -1, -40, 88, -75, -22 };
    qs(array, 0, n - 1);
    for (int i = 0; i < n; i++) printf("%4d", array[i]);
return 0;
}
```

---

# Пірамідальне сортування

<b>Масив до сортування</b>	<b>22</b>	<b>20</b>	<b>-1</b>	<b>-40</b>	<b>88</b>	<b>-75</b>	<b>-22</b>
--------------------------------	-----------	-----------	-----------	------------	-----------	------------	------------

---