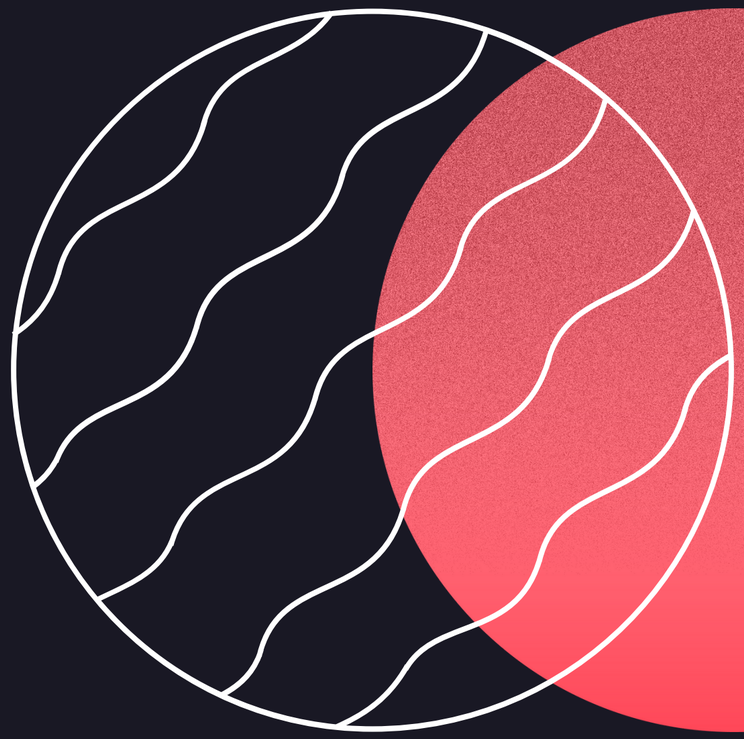
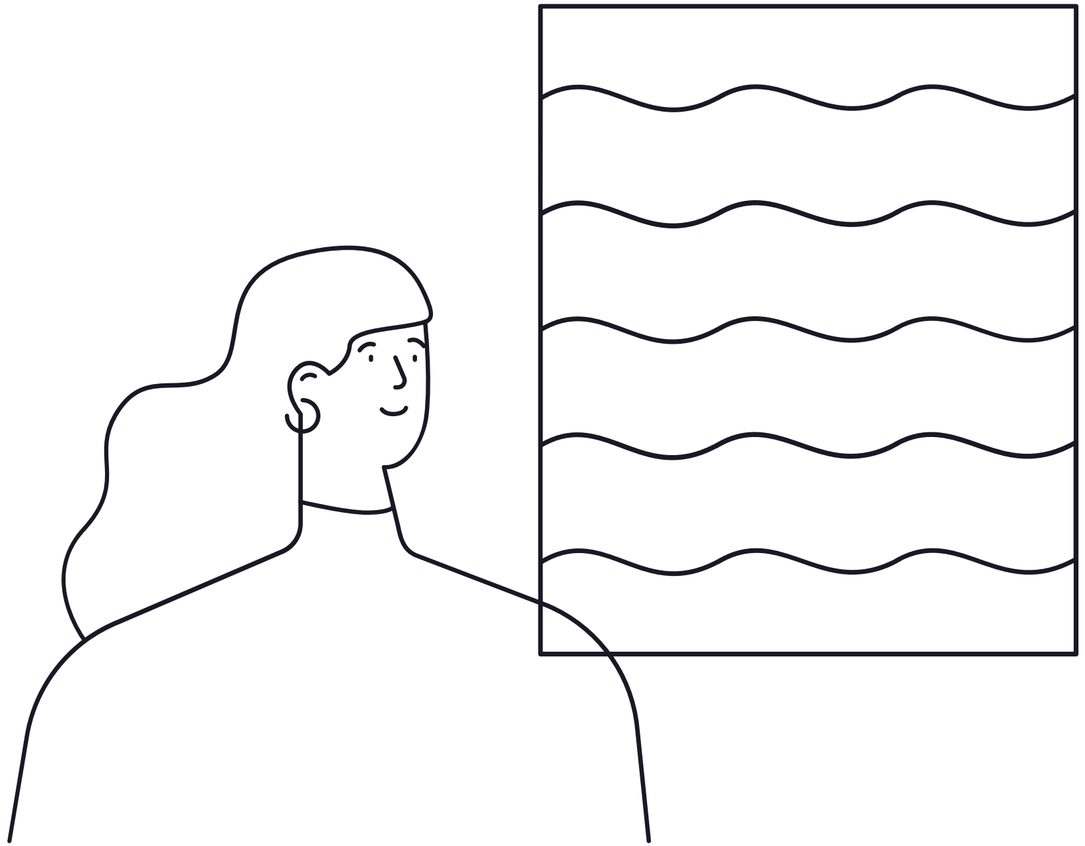
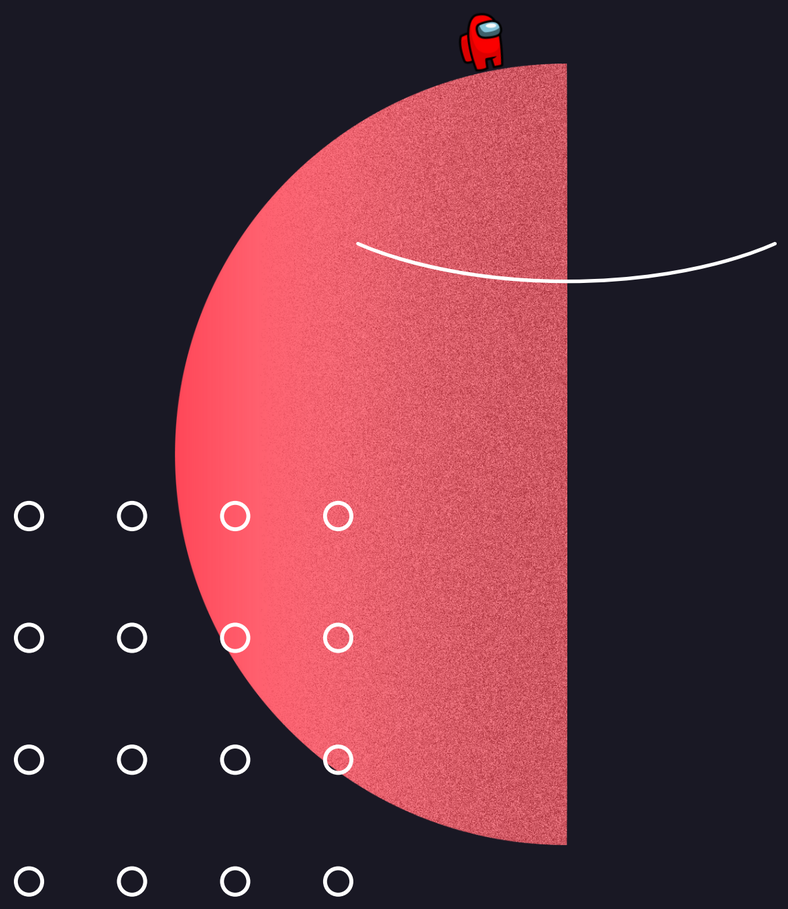


Алгоритми сортування гнома та бульбашкою

Завдання

РЕАЛІЗУВАТИ АЛГОРИТМИ ГНОМА ТА БУЛЬБАШКИ НА МОВАХ ПРОГРАМУВАННЯ C++ ТА PYTHON, ОЦІНИТИ ЇХНЮ СКЛАДНІСТЬ, ТА ПОРІВНЯТИ МІЖ СОБОЮ.



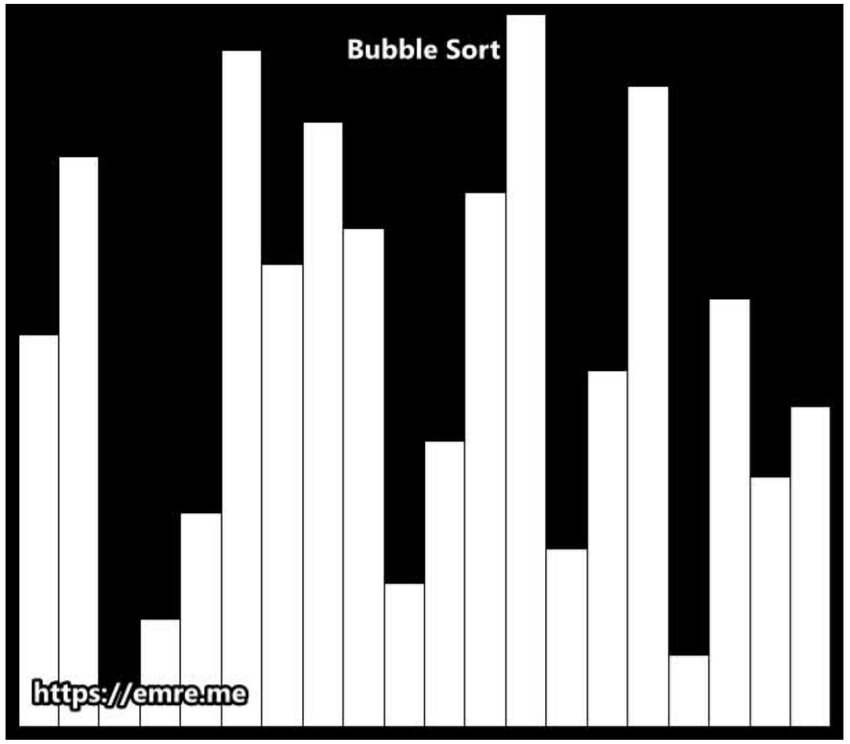


Алгоритми сортування бульбашкою та гнома є простими алгоритмами сортування, які можна використовувати для сортування невеликих наборів даних. Основна різниця між цими алгоритмами полягає у способі переміщення елементів у відсортовану частину масиву.

СОРТУВАННЯ БУЛЬБАШКОЮ

АЛГОРИТМ СОРТУВАННЯ БУЛЬБАШКОЮ

Алгоритм сортування бульбашкою починається з перевірки кожної пари сусідніх елементів і їх обміну, якщо вони не відсортовані. Після першого проходу найбільший елемент знаходиться в кінці масиву. Повторюючи цей процес для решти елементів масиву, вони будуть відсортовані по зростанню. Алгоритм завершується, коли всі елементи відсортовані.



ПРИКЛАД СОРТУВАННЯ БУЛЬБАШКОЮ

Перша ітерація
(9, 5) -> (5, 9) => [5, 9, 2, 7, 1]
(9, 2) -> (2, 9) => [5, 2, 9, 7, 1]
(9, 7) -> (7, 9) => [5, 2, 7, 9, 1]
(9, 1) -> (1, 9) => [5, 2, 7, 1, 9]

Друга ітерація
(5, 2) -> (2, 5) => [2, 5, 7, 1, 9]
(5, 7) -> Без змін => [2, 5, 7, 1, 9]
(7, 1) -> (1, 7) => [2, 5, 1, 7, 9]
(7, 9) -> Без змін => [2, 5, 1, 7, 9]

Третя ітерація
(2, 5) -> Без змін => [2, 5, 1, 7, 9]
(5, 1) -> (1, 5) => [2, 1, 5, 7, 9]
(5, 7) -> Без змін => [2, 1, 5, 7, 9]
(7, 9) -> Без змін => [2, 1, 5, 7, 9]

Четверта ітерація
(2, 1) -> (1, 2) => [1, 2, 5, 7, 9]
(2, 5) -> Без змін => [1, 2, 5, 7, 9]
(5, 7) -> Без змін => [1, 2, 5, 7, 9]
(7, 9) -> Без змін => [1, 2, 5, 7, 9]

ПРОГРАМНА РЕАЛІЗАЦІЯ

C++

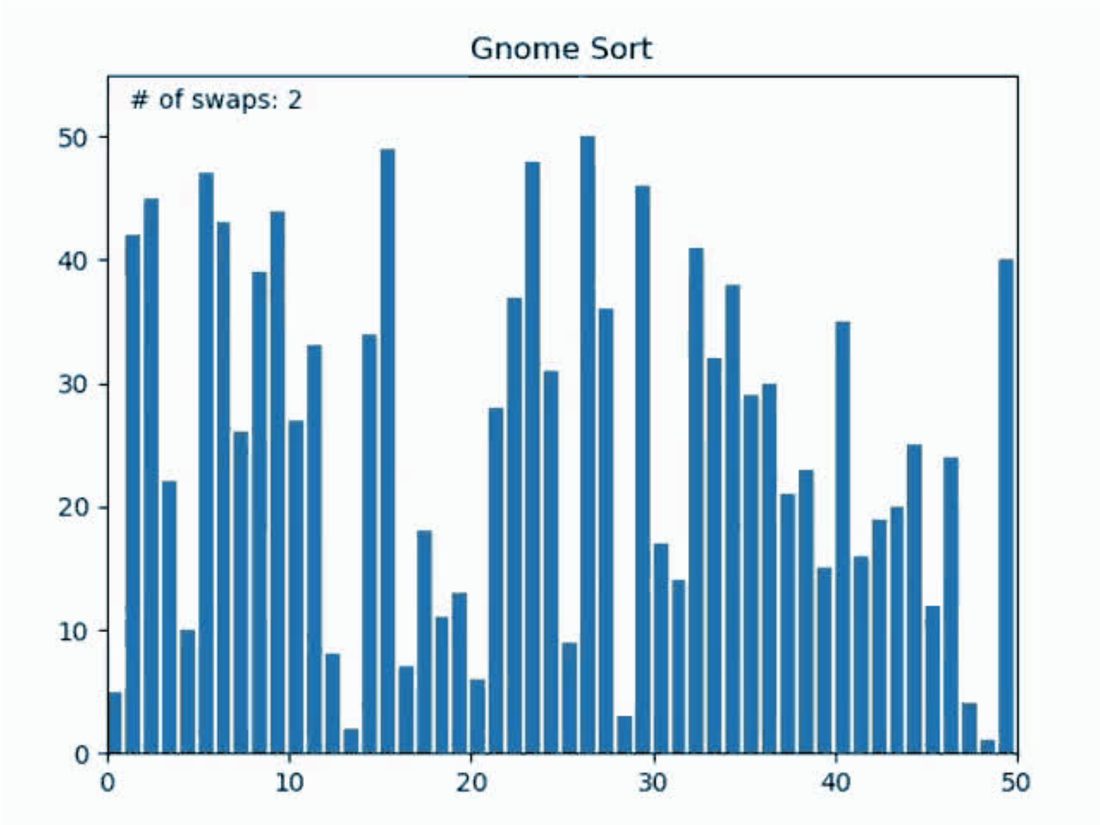
```
void bubble_sort(vector<int>& arr) {
  int n = arr.size();
  for (int i = 0; i < n; i++) {
    bool swapped = false;
    for (int j = 0; j < n - i - 1; j++){
      if (arr[j] > arr[j + 1]) {
        swap(arr[j], arr[j + 1]);
        swapped = true;
      }
    }
    if (!swapped){
      break;
    }
  }
}
```

PYTHON

```
def bubble_sort(arr):
  n = len(arr)
  for i in range(n):
    swapped = False
    for j in range(0, n - i - 1):
      if arr[j] > arr[j + 1]:
        arr[j], arr[j + 1] = arr[j + 1], arr[j]
        swapped = True
    if not swapped:
      break
```

СОРТУВАННЯ ГНОМА

Сортування гнома (gnome sort) є простим алгоритмом сортування, який схожий на сортування вставкою. Алгоритм починає роботу з першого елемента масиву і порівнює його з попереднім елементом, якщо попередній елемент менше поточного, алгоритм переходить до наступного елемента, інакше вони міняються місцями і алгоритм повертається на один елемент назад. Цей процес повторюється дотиди, доки не будуть перевірені всі елементи масиву.



ПРИКЛАД СОРТУВАННЯ ГНОМА

5, 2, 9, 3, 6, 1, 8, 7, 4]
Після 1 ітерації
[2, 5, 9, 3, 6, 1, 8, 7, 4]
Після 2 ітерації
[2, 5, 3, 9, 6, 1, 8, 7, 4]
Після 3 ітерації
[2, 5, 3, 6, 9, 1, 8, 7, 4]
Після 4 ітерації
[2, 5, 3, 6, 1, 9, 8, 7, 4]
Після 5 ітерації
[2, 5, 3, 6, 1, 8, 9, 7, 4]
Після 6 ітерації
[2, 5, 3, 6, 1, 8, 7, 9, 4]
Після 7 ітерації
[2, 5, 3, 6, 1, 8, 7, 4, 9]
Після 7 ітерації
[2, 3, 5, 6, 1, 8, 7, 4, 9]
Після 8 ітерації
[2, 3, 5, 1, 6, 8, 7, 4, 9]
Після 9 ітерації
[2, 3, 1, 5, 6, 8, 7, 4, 9]
Після 10 ітерації
[2, 1, 3, 5, 6, 8, 7, 4, 9]
Після 11 ітерації
[1, 2, 3, 5, 6, 8, 7, 4, 9]
Після 12 ітерації
[1, 2, 3, 5, 6, 7, 8, 4, 9]
Після 13 ітерації
[1, 2, 3, 5, 6, 7, 4, 8, 9]
Після 14 ітерації
[1, 2, 3, 5, 6, 4, 7, 8, 9]
Після 15 ітерації
[1, 2, 3, 5, 4, 6, 7, 8, 9]
Після 16 ітерації
[1, 2, 3, 4, 5, 6, 7, 8, 9]

Посортований
масив: [1, 2, 3, 4, 5, 6, 7, 8, 9]

ПРОГРАМНА РЕАЛІЗАЦІЯ

C++

```
vector<int> gnome_sort(vector<int> arr)
{
  int n = arr.size();
  int index = 0;
  while (index < n) {
    if (index == 0){
      index++;
    }
    if (arr[index] >= arr[index - 1]){
      index++;
    }
    else {
      swap(arr[index], arr[index - 1]);
      index--;
    }
  }
  return arr;
}
```

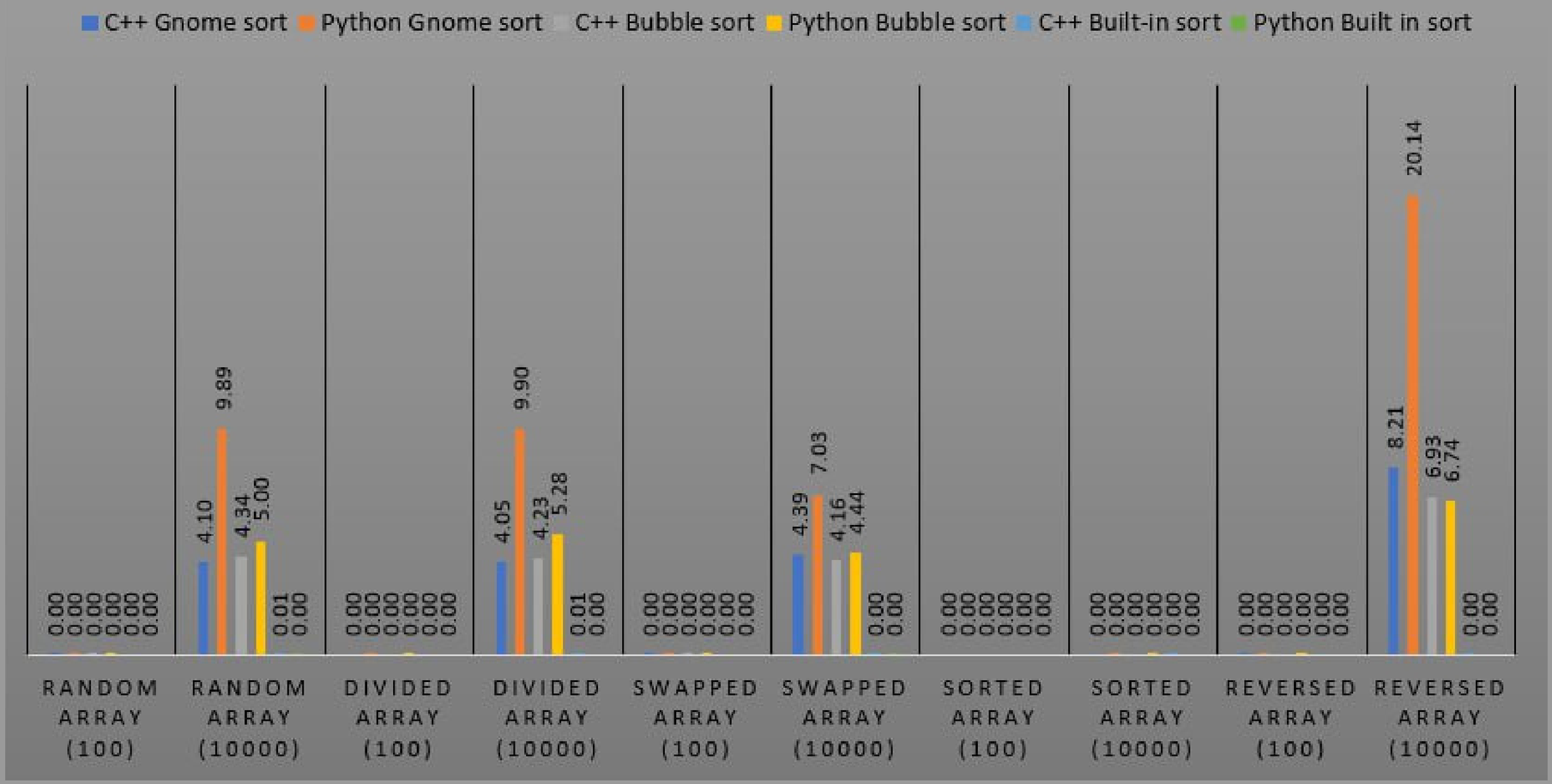
PYTHON

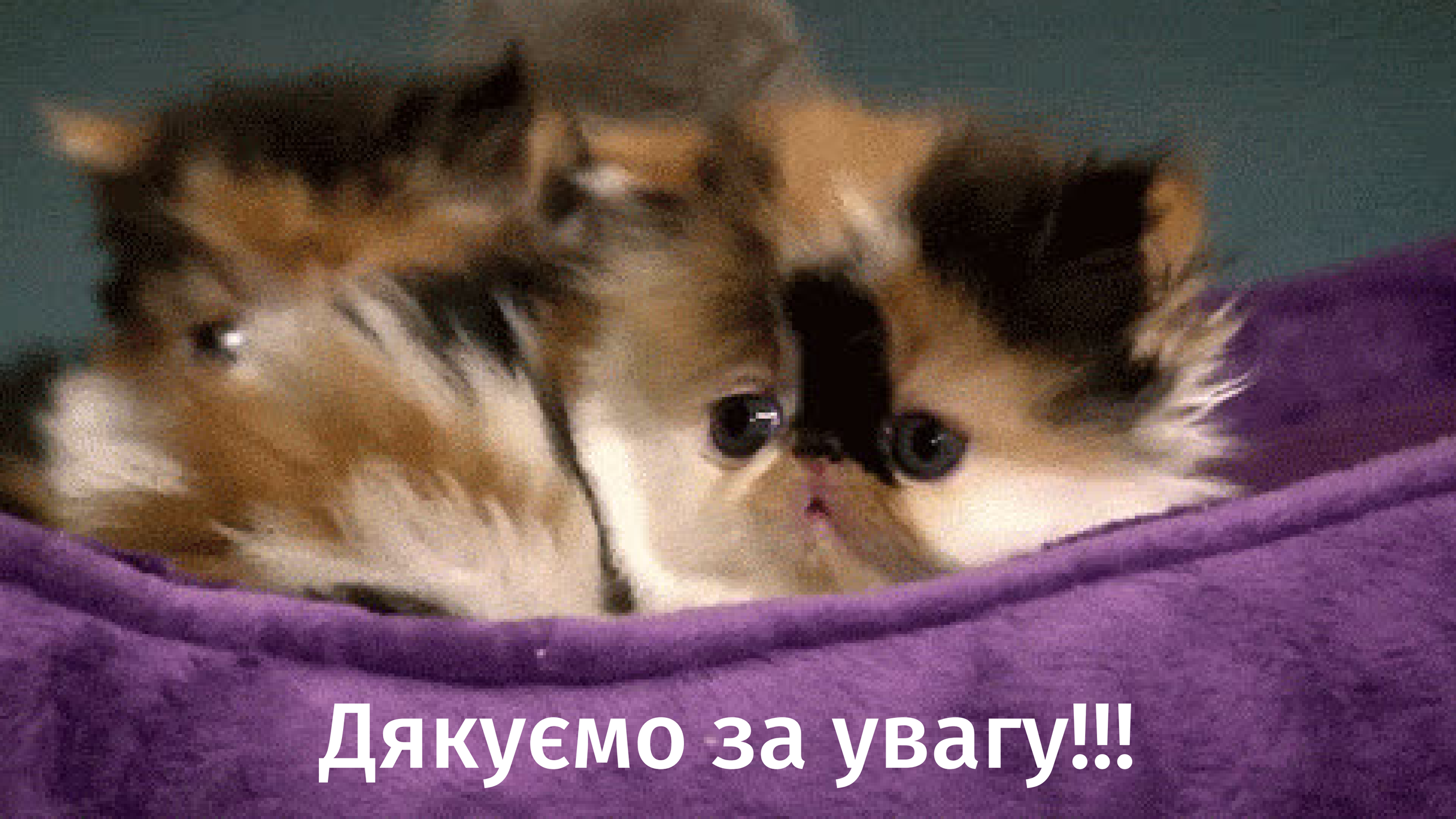
```
def gnome_sort(arr):
  n = len(arr)
  index = 0
  while index < n:
    if index == 0:
      index = index + 1
    if arr[index] >= arr[index - 1]:
      index = index + 1
    else:
      arr[index], arr[index - 1] = arr[index - 1],
arr[index]
      index = index - 1
  return arr
```

Порівняльна таблиця алгоритмів

	Random Array (100)	Random Array (10000)	Divided Array (100)	Divided Array (10000)	Swapped Array (100)	Swapped Array (10000)	Sorted Array (100)	Sorted Array (10000)	Reversed Array (100)	Reversed Array (10000)
C++ Gnome sort	0.00	4.10	0.00	4.05	0.00	4.39	0.00	0.00	0.00	8.21
Python Gnome sort	0.00	9.89	0.00	9.90	0.00	7.03	0.00	0.00	0.00	20.14
C++ Bubble sort	0.00	4.34	0.00	4.23	0.00	4.16	0.00	0.00	0.00	6.93
Python Bubble sort	0.00	5.00	0.00	5.28	0.00	4.44	0.00	0.00	0.00	6.74
C++ Built-in sort	0.00	0.01	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00
Python Built in sort	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Порівняльна гістограма алгоритмів





Дякуємо за увагу!!!