

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»
НАВЧАЛЬНО-НАУКОВИЙ ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ**

Лабораторна робота № 2
з дисципліни «Алгоритми та структури даних»
На тему: «Методи сортування масивів»

Виконав:
студент групи ФІ-12
Завалій Олександр

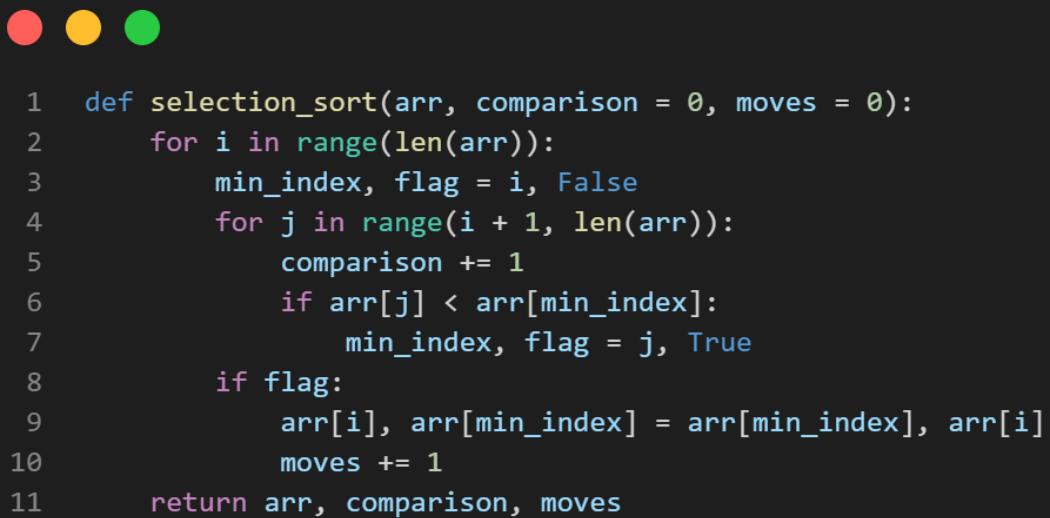
Реалізація завдання

Task I

Варіант №5

Написати програму, що реалізує один з простих методів сортування. Сортування методом вибору.

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



```
1 def selection_sort(arr, comparison = 0, moves = 0):
2     for i in range(len(arr)):
3         min_index, flag = i, False
4         for j in range(i + 1, len(arr)):
5             comparison += 1
6             if arr[j] < arr[min_index]:
7                 min_index, flag = j, True
8         if flag:
9             arr[i], arr[min_index] = arr[min_index], arr[i]
10            moves += 1
11    return arr, comparison, moves
```

Результати виконання алгоритму.

```
Selection sort
Comparison: 4950. Moves: 94
Comparison: 499500. Moves: 994
Comparison: 49995000. Moves: 9988

Execution time
Array: 100. Algorithm: selection_sort. Minimum execution time: 0.004801600007340312
Array: 1000. Algorithm: selection_sort. Minimum execution time: 0.4833672000095248
Array: 10000. Algorithm: selection_sort. Minimum execution time: 49.92777469998691
```

Task II

Написати програму, що реалізує метод швидкого сортування.

$\sim n \log_2 n$

```
1  comparison1 = 0
2  def quick_sort(arr):
3      global comparison1
4      low, same, high = [], [], []
5
6      if len(arr) > 1:
7          pivot = choice(arr)
8          for x in arr:
9              comparison1 += 1
10             if x < pivot:
11                 low.append(x)
12             elif x == pivot:
13                 same.append(x)
14             elif x > pivot:
15                 high.append(x)
16         return quick_sort(low)+same+quick_sort(high)
17     else:
18         return arr
```

Результати виконання алгоритму.

```
Quick sort
Comparison, moves: 1383
Comparison, moves: 11384
Comparison, moves: 173538

Execution time
Array: 100. Algorithm: quick_sort. Minimum execution time: 0.0016931999707594514
Array: 1000. Algorithm: quick_sort. Minimum execution time: 0.024525300017558038
Array: 10000. Algorithm: quick_sort. Minimum execution time: 0.3424866999848746
```

Функція, що використовується для підрахунку часу виконання іншого блоку кода.

```
1 def run_time(algorithm, array):
2     setup_code = f"from __main__ import {algorithm}"
3     stmt = f"{algorithm}({array})"
4     times = repeat(setup=setup_code, stmt=stmt, repeat=3, number=10)
5     print(f"Array: {len(array)}. Algorithm: {algorithm}. Minimum execution time: {min(times)}")
```

Початковий та відсортовані масиви.

```
Initial array:
[-8, -141, 73, 284, -992, -77, -276, -765, 110, -122, 185, -662, 937, 325, -79, 225, -723, -894, -212,
-416, 688, 72, -217, -936, -183, -447, -549, -706, 273, 5, -165, -143, -398, -903, -312, -828, -327,
-778, -656, 287, 559, 703, -45, 412, 52, -913, 122, 466, 403, 814, -144, 52, -784, -487, 152, 83, 150,
720, -10, -556, -20, 788, 570, 236, -617, -325, 504, 145, -919, 721, -25, -520, 271, -529, -291, -352,
685, 799, -438, 609, 384, -156, -679, 860, 175, 975, 79, 686, 1000, 512, 184, -792, 944, -677, 744,
811, 168, -663, -948, -287]

Selection sort:
[-992, -948, -936, -919, -913, -903, -894, -828, -792, -784, -778, -765, -723, -706, -679, -677, -663,
-662, -656, -617, -556, -549, -529, -520, -487, -447, -438, -416, -398, -352, -327, -325, -312, -291,
-287, -276, -217, -212, -183, -165, -156, -144, -143, -141, -122, -79, -77, -45, -25, -20, -10, -8, 5
, 52, 52, 72, 73, 79, 83, 110, 122, 145, 150, 152, 168, 175, 184, 185, 225, 236, 271, 273, 284, 287, 3
25, 384, 403, 412, 466, 504, 512, 559, 570, 609, 685, 686, 688, 703, 720, 721, 744, 788, 799, 811, 814
, 860, 937, 944, 975, 1000]

Quick sort:
[-992, -948, -936, -919, -913, -903, -894, -828, -792, -784, -778, -765, -723, -706, -679, -677, -663,
-662, -656, -617, -556, -549, -529, -520, -487, -447, -438, -416, -398, -352, -327, -325, -312, -291,
-287, -276, -217, -212, -183, -165, -156, -144, -143, -141, -122, -79, -77, -45, -25, -20, -10, -8, 5
, 52, 52, 72, 73, 79, 83, 110, 122, 145, 150, 152, 168, 175, 184, 185, 225, 236, 271, 273, 284, 287, 3
25, 384, 403, 412, 466, 504, 512, 559, 570, 609, 685, 686, 688, 703, 720, 721, 744, 788, 799, 811, 814
, 860, 937, 944, 975, 1000]
```

Результати порівняння методів сортування

	Сортування методом вибору					Швидке сортування				
N	К-ть копіювань		К-ть порівнянь		Час	К-ть копіювань		К-ть порівнянь		Час
	(M)		(C)			(M)		(C)		
	Теорет.	Експерим.	Теорет.	Експерим.	(T)	Теорет.	Експерим.	Теорет.	Експерим.	(T)
100	99	93	4950	4950	0.0047	664	687	664	687	0.00162
1000	999	996	499500	499500	0.4166	9966	11371	9966	11371	0.02219
10000	9999	9992	49995000	49995000	43.4776	132877	160178	132877	160178	0.2965