

Упражнение 2.

Бавни сортираниз. Стабилен/In-place.

1. Стабилност - ако една сортировка ни гарантира, че елементите, които са равни един спрямо друг, няма да си променят позициите \rightarrow стабилен.

Пример 1:

Сашо - 45 точки

Ели - 50 точки

Гери - 60 точки

Иван - 50 точки

Никола - 60 точки

Мария - 50 точки

сортираме

1. Гери - 60 точки

2. Никола - 60 точки

3. Ели - 50 точки

4. Иван - 50 точки

5. Мария - 50 точки

6. Сашо - 45 точки

Пример 2: (3,5) (2,1) (3,0) \rightarrow (3,5) (3,0) (2,1)

2. In-place - да не се използва допълнителна памет за сортирането на елементите.

Bubble sort - възходящ ред

Пример:

5	4	2	6	3
4	5	2	6	3
4	2	5	6	3
4	2	5	3	6

\Rightarrow

След първата итерация \rightarrow
максималният елемент е
"избутан" до мястото си

4	2	5	3	6
2	4	5	3	6
2	4	5	3	6
2	4	3	5	6

- няма разлика

Bubble Sort

стабилност	✓
in-place	✓
сложност	$O(n^2)$

- разменяване ел. \Rightarrow удовлетворява $\leq / >$
- размесиване си в масива. ^^

Selection Sort - възходящ ред

На всяка стъпка поставяме текущия min ел. на мястото му.

стабилност	✗
in-place	✓
сложност	$O(n^2)$

Контрпример за стабилността: /ст = стъпка/

4A 5 3 2 4B 1

ст. 1: 1 5 3 2 4B 4A

ст. 2: 1 2 3 5 4B 4A

ст. 3: 1 2 3 4B 5 4A

ст. 4: 1 2 3 4B 4A 5

- прави най-малко разменения на елементите

Insertion sort - възходящ ред

/Сортиране чрез вмъкване/

Всички елементи до i -та позиция са сортирани

Пример:

5 | 4 2 6 3

↙

сортирано

Анализ Insertion sort

стабилност	✓
in-place	✓
сложност	$O(n^2)$

ст. 1: 4 5 | 2 6 3

ст. 2: 2 4 5 | 6 3

ст. 3: 2 4 5 6 | 3

ст. 4: 2 3 4 5 6

Използване "стабилност": разпознаваме ел. само за " $<$ " или " $>$ ", т.е. равни ел. не ни интересуват