

Анализа алгоритма selection sort

Александар Спасић

Фебруар 2026.

Садржај

1 Принцип рада selection sort

1.1 Увод
1.1.1 Временска сложеност
1.2 Закључак

Глава 1

Принцип рада selection sort

1.1 Увод

Алгоритам *selection sort* је један од најосновнијих алгоритама сортирања. Базира се на следећем принципу: ако низ има више од једног елемента, замени почетни елемент са најмањим елементом низа и затим аналогно сортирај остатак низа (елементе иза почетног). У свакој итерацији се на своју позицију доводи следећи по величини елемент низа, тј. у i -тој итерацији се i -ти по величини елемент доводи на позицију i . Ово се може реализовати тако што се пронађе позиција m најмањег елемента од позиције i до краја низа и затим се размене елемент на позицији i и елемент на позицији m . Алгоритам се зауставља када се претпоследњи по величини елемент доведе на претпоследњу позицију у низу.

Пример:

- Прикажимо рад алгоритма на примеру сортирања низа 2 12 10 15 20.
 1. .2 12 10 15 20, $i = 0, m = 4$, размена елемената 2 и 2.
 2. 2 .12 10 15 20, $i = 1, m = 2$, размена елемената 12 и 10.
 3. 2 10 .12 15 20, $i = 2, m = 2$, размена елемената 10 и 12.

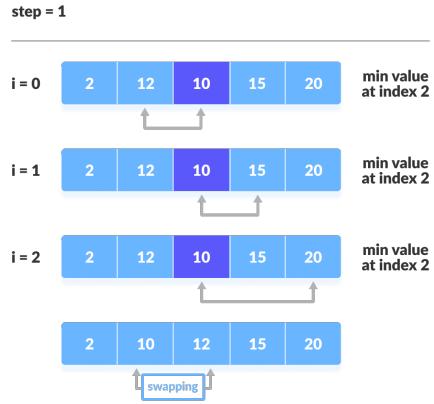
Визуелни пример рада је приказан на слици 1.1.

1.1.1 Временска сложеност

Теорема 1.1. Временска сложеност selection sort алгоритма је

$$O(n) = n^2 \tag{1.1}$$

Значи, овај алгоритам је врло неефикасан. У табели 1.1 је приказана сложеност алгоритма selection sort у односу на друге алгоритме сортирања:



Слика 1.1: Приказ рада алгоритма selection sort.

Алгоритам	Временска сложеност
Selection sort	$O(n) = n^2$
Quick sort	$O(n) = n^2$
Bubble sort	$O(n) = n^2$
Heap sort	$O(n) = n \log n$
Merge sort	$O(n) = n \log n$

Табела 1.1: Приказ временских сложености различитих алгоритама.

1.2 Закључак

И онако алгоритам сортирања selection sort је врло неефикасан, он је врло важан да почетницима покаже елементарне технике писања алгоритама.