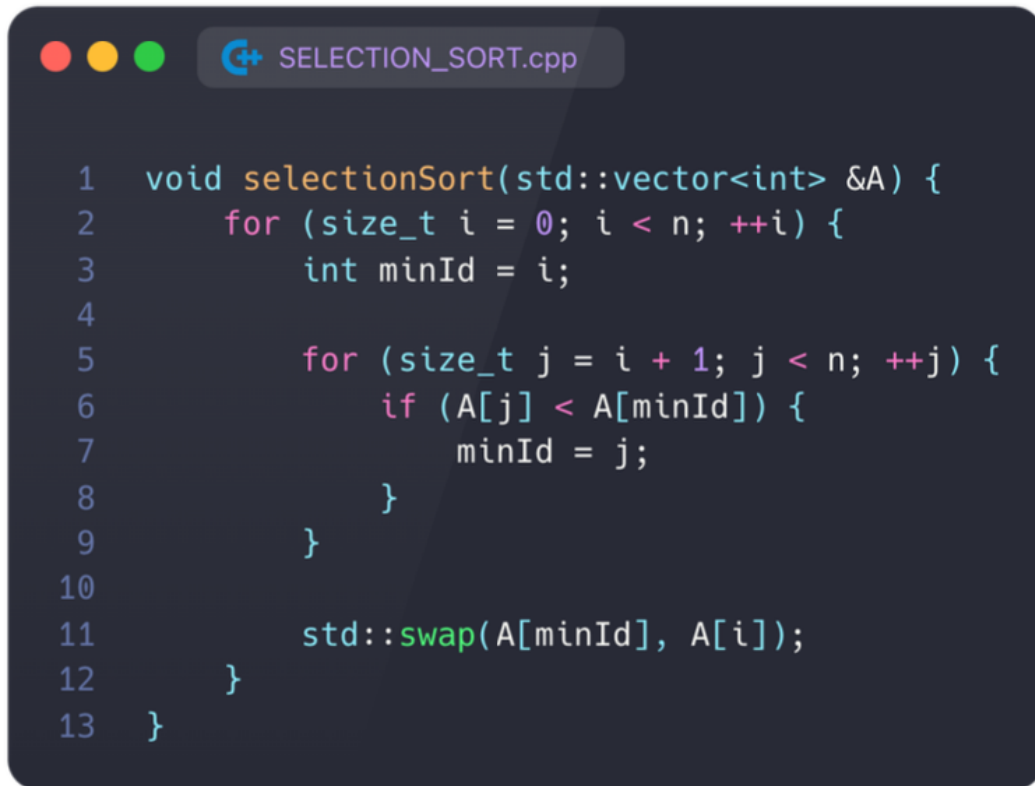


Задача А1. Анализ корректности SELECTION SORT

Дана следующая итерационная реализация алгоритма сортировки выбором заданного целочисленного массива A , количество элементов в котором обозначено n :



```
1 void selectionSort(std::vector<int> &A) {
2     for (size_t i = 0; i < n; ++i) {
3         int minId = i;
4
5         for (size_t j = i + 1; j < n; ++j) {
6             if (A[j] < A[minId]) {
7                 minId = j;
8             }
9         }
10
11         std::swap(A[minId], A[i]);
12     }
13 }
```

1. Сформулируйте условие $P1$, которое подходит в качестве инварианта внутреннего цикла алгоритма по j . Представьте краткое обоснование (например, с использованием частичной трассировки выполнения цикла).

Инвариант $P1$:

На начале каждой итерации внутреннего цикла (по j), переменная $minId$ содержит индекс минимального элемента в подмассиве $A[i...j-1]$. То есть, для всех k от i до $j-1$ выполняется: $A[minId] \leq A[k]$.

Обоснование:

INIT: Перед началом цикла j , $minId = i$. Подмассив $A[i...i-1]$ пустой, поэтому инвариант выполняется тривиально.

MNT: На каждой итерации, если $A[j] < A[minId]$, обновляем $minId = j$, иначе $minId$ не изменяется. Таким образом, после каждой итерации $minId$ указывает на минимальный элемент в $A[i...j]$.

TRM: После выхода из цикла по j , $minId$ указывает на индекс минимального элемента в $A[i..n-1]$, что необходимо для корректного выполнения обмена $swap(A[minId], A[i])$.

2. Сформулируйте условие P2, которое подходит в качестве инварианта внешнего цикла алгоритма по i. Представить краткое обоснование.

Инвариант P2:

На начале каждой итерации внешнего цикла (по i), подмассив $A[0...i-1]$ отсортирован по возрастанию, и каждый его элемент меньше либо равен всем элементам подмассива $A[i...n-1]$. То есть:

Для всех k от 0 до i-2 выполняется: $A[k] \leq A[k+1]$.

Для всех m от i до n-1 и всех k от 0 до i-1 выполняется: $A[k] \leq A[m]$.

Обоснование:

INIT: Перед первым проходом (когда $i = 0$), подмассив $A[0...-1]$ пустой, поэтому инвариант выполняется тривиально.

MNT: На каждой итерации мы находим минимальный элемент в $A[i...n-1]$ и ставим его на позицию i. Таким образом, после обмена, $A[0...i]$ содержит все минимальные элементы в отсортированном порядке.

TRM: После выхода из внешнего цикла (когда $i = n$), весь массив $A[0...n-1]$ отсортирован по возрастанию.

3. Выполните проверку выполнения найденных инвариантов P1 и P2 до входа в каждый из циклов (INIT), в каждой итерации циклов (MNT), при выходе из цикла (TRM).

Для инварианта P1 (внутренний цикл по j):

INIT: Перед входом в цикл по j, $\text{minId} = i$, $j = i + 1$. Подмассив $A[i...j-1]$ состоит из одного элемента $A[i]$. Выполняется $A[\text{minId}] = A[i] \leq A[i]$, инвариант соблюдается.

MNT: Предположим, инвариант выполняется в начале итерации. После проверки $\text{if } (A[j] < A[\text{minId}])$, возможны два случая: Если $A[j] < A[\text{minId}]$, то $\text{minId} = j$. Теперь $A[\text{minId}]$ — минимальный в $A[i...j]$. Иначе minId не изменяется, и минимальный элемент остается тем же. В обоих случаях инвариант сохраняется.

TRM: После выхода из цикла по j, minId указывает на минимальный элемент в $A[i...n-1]$. Инвариант P1 обеспечивает корректность дальнейшего обмена элементов.

Для инварианта P2 (внешний цикл по i):

INIT: Перед входом в цикл по i, $i = 0$. Подмассив $A[0...-1]$ пустой, инвариант выполняется тривиально.

MNT: Предполагая, что $A[0...i-1]$ отсортирован и каждый его элемент \leq всех элементов в $A[i...n-1]$. После обмена $\text{swap}(A[\text{minId}], A[i])$, минимальный элемент из $A[i...n-1]$ перемещается на позицию i. Теперь $A[0...i]$ отсортирован, и каждый его элемент \leq всех последующих. Инвариант сохраняется.

TRM: После завершения цикла по i (когда $i = n$), весь массив $A[0..n-1]$ отсортирован по возрастанию. Инвариант P2 гарантирует полную сортировку массива.

Таким образом, инварианты P1 и P2 выполняются на всех этапах (INIT, MNT, TRM), что подтверждает корректность алгоритма сортировки выбором.