

# A1. Анализ корректности SELECTION SORT

Демченко Георгий Павлович , БПИ-235

SELECTION\_SORT.cpp

```
void selectionSort(std::vector<int> &A) {
    //INIT P2
    for (size_t i = 0; i < n; ++i) {    //MNT P2
        int minId = i;    //INIT P1

        for (size_t j = i + 1; j < n; ++j) {    //MNT P1
            if (A[j] < A[minId]) {
                minId = j;
            }
        }
        //TRM P1
        std::swap(A[minId], A[i])
    }
    //TRM P2
}
```

## 1. Инвариант внутреннего цикла $P_1$

- $P_1$  : В начале каждой  $j$ -ой итерации **minId** - индекс минимального элемента в срезе массива  $A - A[i; j - 1]$
- **Краткое обоснование:** На каждой  $j$ -ой итерации ( $j \in [i + 1; n - 1]$ ) сравним текущий элемент  $A[j]$  с минимальным  $A[\text{minId}]$ , где **minId**  $\in [i; j - 1]$  и обновляем minId на  $j$  если  $A[j] < A[\text{minId}]$ . Получаем **minId** - индекс минимального элемента среди  $A[i; j]$ . При условии верной базы  $\text{minId} = i$  - индекс минимального элемента в  $A[i; i]$  при  $j = i + 1$ , это утверждение верно на каждой итерации.

## 2. Инвариант внешнего цикла $P_2$

- $P_2$  : После каждой  $i$ -ой итерации часть массива  $A - A[0 : i]$  отсортирована по неубыванию
- **Краткое обоснование:** На каждой  $i$ -ой ( $i \in [0; n - 1]$ ) итерации находим minId - индекс минимального элемента в  $A[i; n - 1]$  и меняем местами  $A[i]$  и  $A[\text{minId}]$ , тем самым образуя неубывающую последовательность  $A[0; i]$  из минимальных элементов массива

## 3. Проверка выполнения $P_1, P_2$

### 3.1: $P_1$

- **INIT:**  $j = i + 1 \Rightarrow A[i; j - 1] = A[i; i]$  - содержит один элемент  $\Rightarrow \text{minId} = i$  - индекс минимального элемента

- **MNT:** На каждой итерации по  $j \in [i + 1; n - 1]$  сравниваем текущий  $A[j]$  элемент с минимальным элементом  $A[\text{minId}]$ ,  $\text{minId} \in [i; j - 1]$  и обновляем **minId** если  $A[j] < A[\text{minId}] \Rightarrow \text{minId}$  - индекс минимального элемента из  $A[i; j]$ . При корректной базе (**INIT**) получаем, что на входе в каждую  $j$ -ую итерацию **minId** - индекс минимального элемента среди  $A[i; j - 1]$
- **TRM:**  $j = n \Rightarrow A[i; j - 1] = A[i; n - 1]$  - в **MNT** перебрали все элементы  $A[i; n - 1]$  и нашли минимальный, корректно установив **minId** - минимальный индекс элемента среди  $A[i; n - 1]$

### 3.2 : $P_2$

- **INIT:** Итераций не выполнялось,  $i = 0 \Rightarrow A[0; i] = A[0; 0]$  - содержит один элемент  $\Rightarrow$  отсортирован по неубыванию.
- **MNT:** На каждой итерации по  $i$  ( $i \in [0; n - 1]$ ) находим **minId** - индекс минимального элемента в  $A[i; n - 1]$  в цикле по  $j$ , меняем местами  $A[i]$  и  $A[\text{minId}]$ , тем самым образуя неубывающую последовательность  $A[0; i]$  из минимальных элементов массива
- **TRM:**  $i = n$  - итерация не выполняется, последняя выполненная итерация  $i = n - 1 \Rightarrow$  согласно **MNT** массив  $A[0; n - 1]$  - весь массив отсортирован по неубыванию