1. P1: На каждой итерации minId остается равным индексу минимального элемента на A[i:j+1].
2. P2: На каждой итерации массив A[0:i] остается отсортированным по неубыванию. Так как A[i] либо остается на месте, либо меняется на меньший элемент из стоящих впереди него.
3. Проверка для P1:

INIT: очевидно, что элемент с индексом i является наименьшим в A[i:i+1]

MNT: на каждой итерации программа сравнивает текущий элемент с наименьшим, если текущий элемент меньше – заменяет индекс прошлого наименьшего на индекс текущего.

TRM: перед выходом из цикла, последний элемент сравнивается с наименьшим и, при необходимости, заменяет его. После выхода из цикла остаемся с переменной minId, которая равна индексу минимального элемента в массиве A[i:j+1]

Проверка для P2:

INIT: до входа в цикл пустой массив является отсортированным

MNT: на каждой итерации A[i] меняется с наименьшим из оставшихся чисел, либо остается на месте, если уже является наименьшим. Очевидно, что каждое новое наименьшое число будет не меньше предыдущего, а следовательно на каждой итерации A[0:i] будет отсортирован по неубыванию.

TRM: перед выходом из цикла мы поставим последнее число на его место, и останемся с отсортированным по неубыванию массивом A