

Лектор: А.Д.Хапугин

# Основы программного конструирования

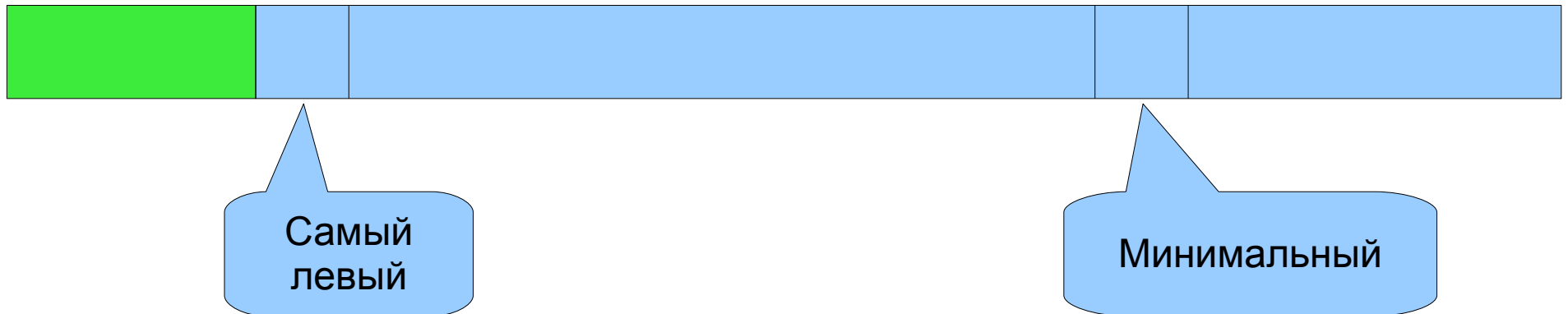
Лекция 9-1. Сортировки (продолжение)

Материалы доступны в Интернете по адресу: <http://www.excelsior.ru/afti/>

# Простой («прямой») выбор

Выбираем самый маленький элемент из массива и ставим его на первое место.  
Выбираем самый маленький элемент из того, что осталось, и ставим его на второе место.  
Продолжаем процесс пока не дойдем до конца.

Чтобы реализовать, надо уточнить: что значит «ставим его на «энное» место?»



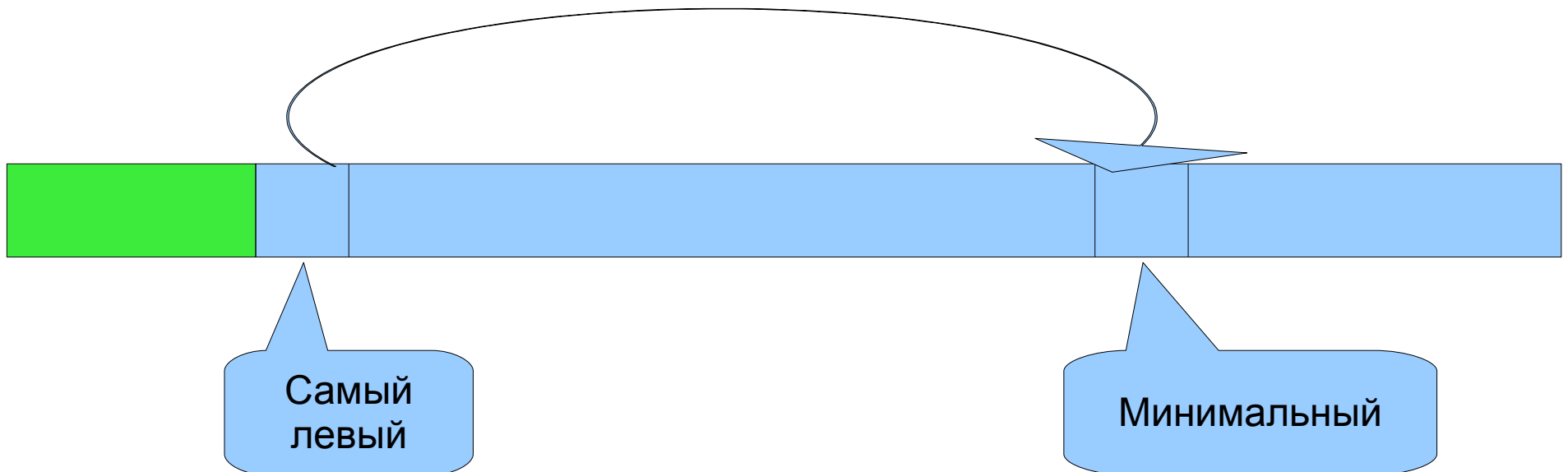
# Уточняем алгоритм

Вводим понятия:

- неотсортированная часть массива (в начале работы — весь массив)
- отсортированная часть массива (в начале работы — пусто)

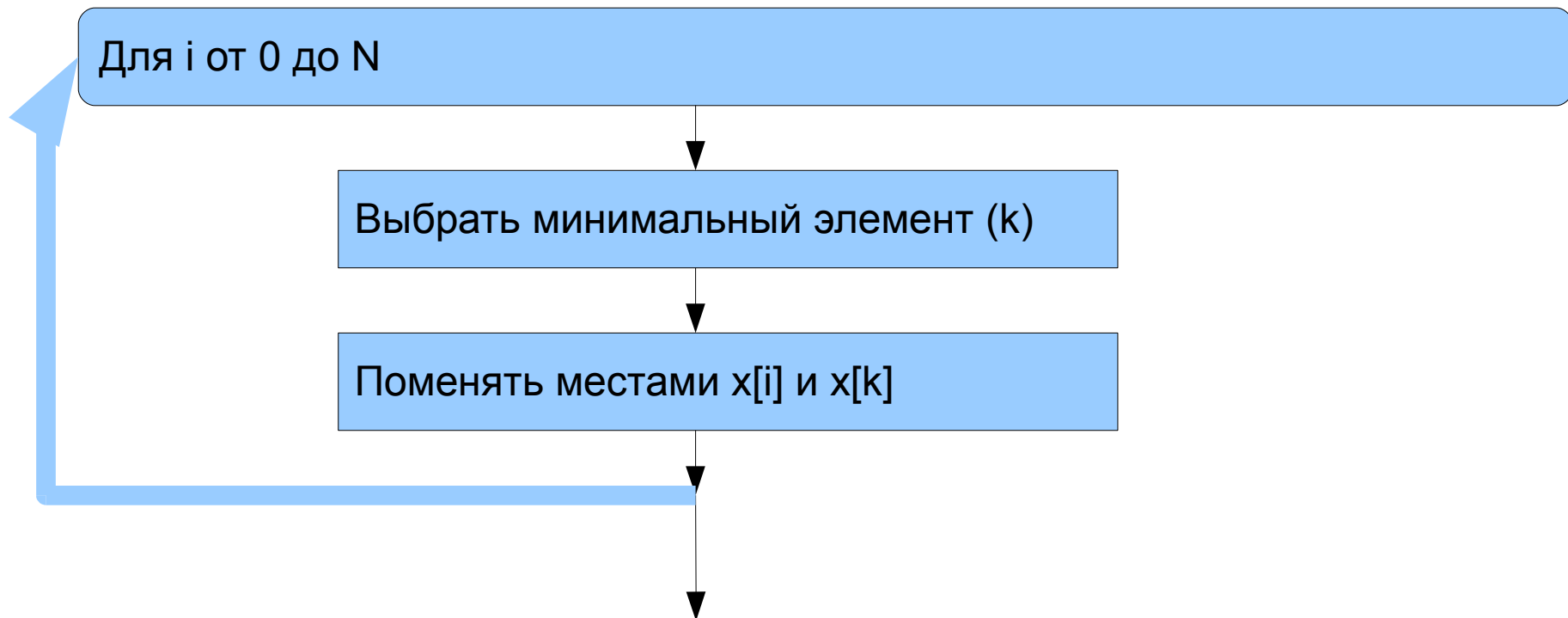
Алгоритм:

- 1) выбираем минимальный элемент из неотсортированной части массива
- 2) меняем его местами с самым левым элементом неотсортированной части массива
- 3) сдвигаем границу отсортированной части на 1 вправо
- 4) продолжаем шаги 1-3 до исчерпания неотсортированной части

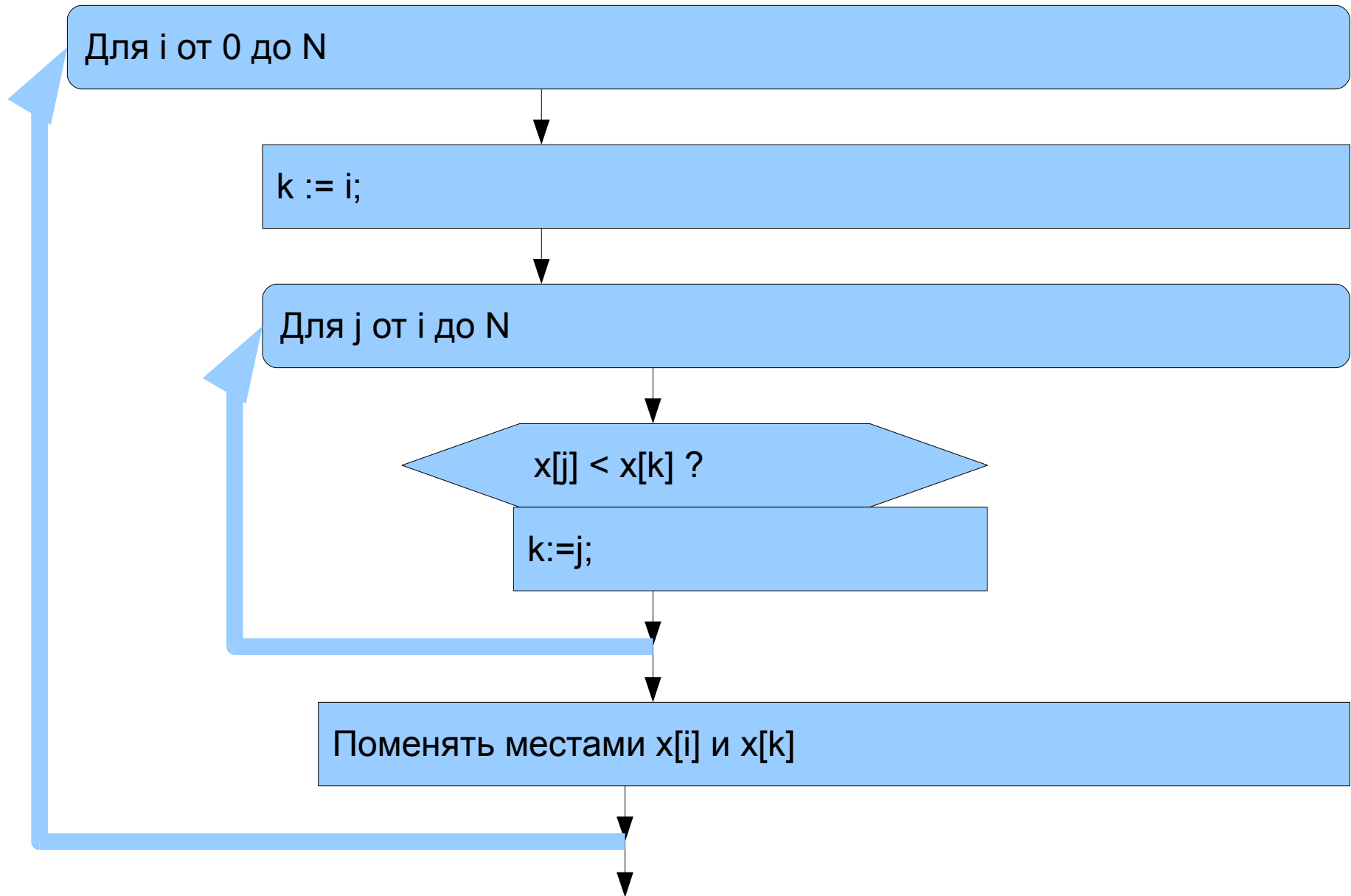


# Блок-схема

$i$  — самый левый элемент неотсортированной части массива



# Уточнение блок-схемы



# Контрольные вопросы

Сложность алгоритма?

$N^2/2$

Какой случай наихудший?

Нет такого

Какой случай наилучший?

Нет такого

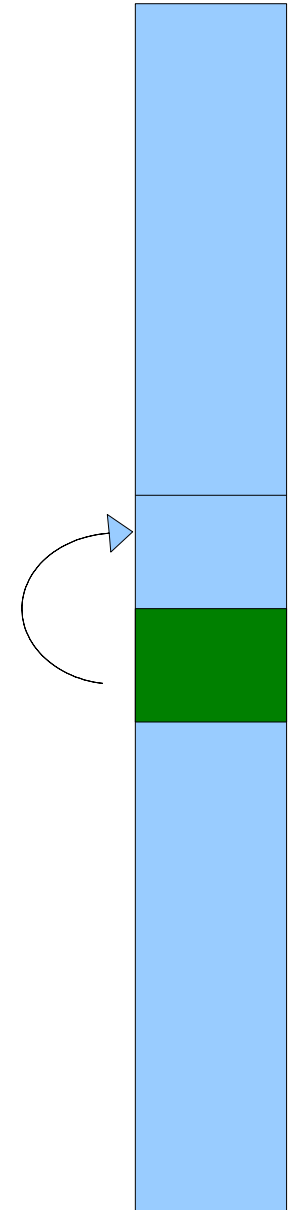
# Пузырек

Идея алгоритма: объединяем проверки и перестановки.

- 1) сравниваем 2 соседних элемента.
- 2) если порядок изменяется, меняем их местами
- 3) сдвигаемся на 1 шаг.

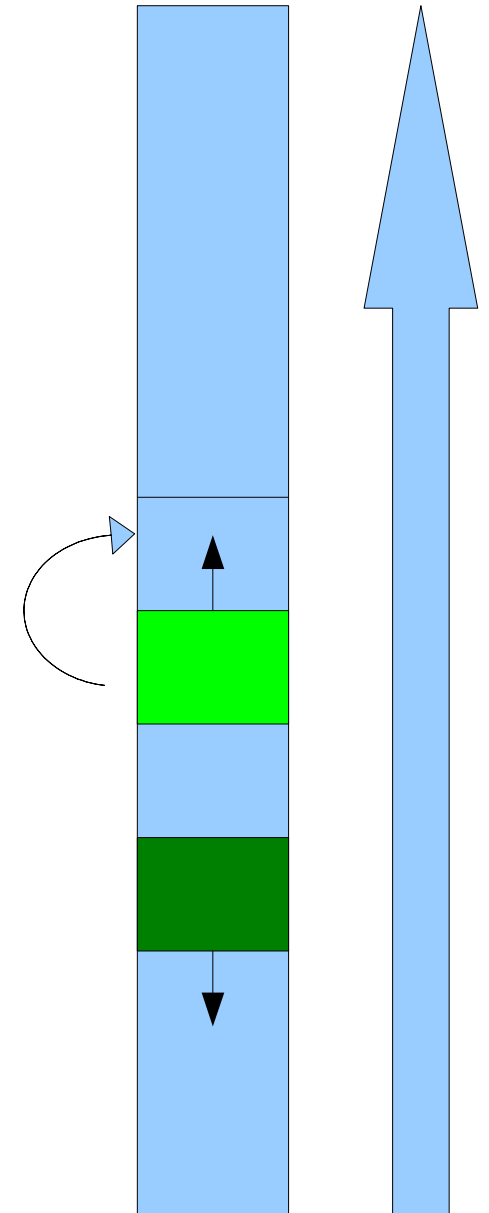
Почему пузырек?

Завершение?



# Алгоритм

Двигаемся слева направо  
если встречаем нарушение порядка, меняем элементы местами  
Повторяем итерации, если была хоть одна перестановка





# Идея усовершенствования

Вопросы:

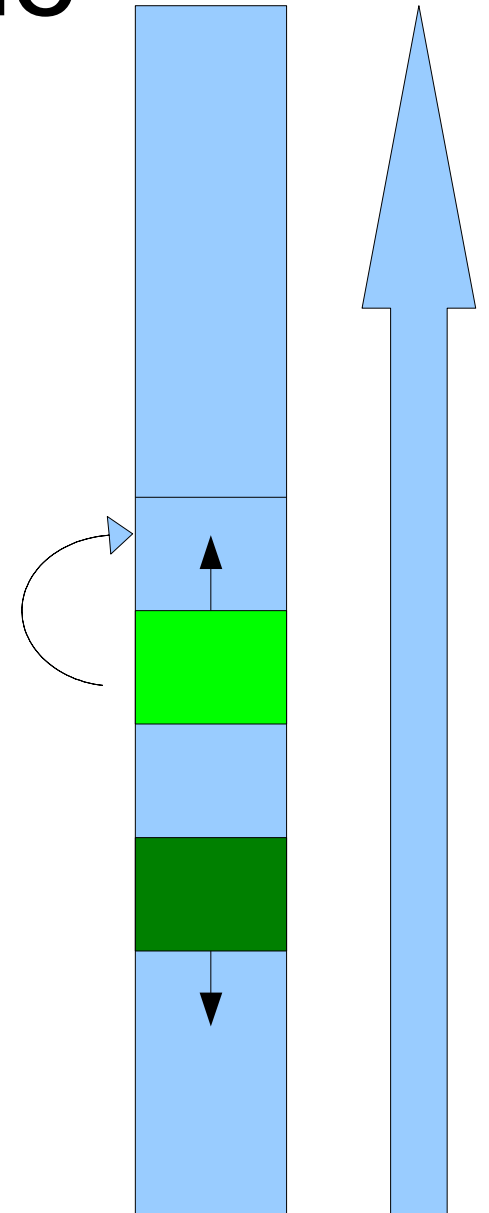
1. За сколько проходов самый «легкий» элемент поднимется наверх?
2. За сколько проходов самый «тяжелый» элемент опустится в самый низ?
3. Как это использовать?

# Усовершенствование

Двигаемся слева направо, если встречаем нарушение порядка, меняем элементы местами

Двигаемся справа налево, если встречаем нарушение порядка, меняем элементы местами

Повторяем итерации, если была хоть одна перестановка хоть в одном из проходов слева направо или справа налево



# Еще идея

После первого прохода вверх, где окажется самый «легкий» элемент?

После первого прохода вверх, где окажется самый «тяжелый» элемент?

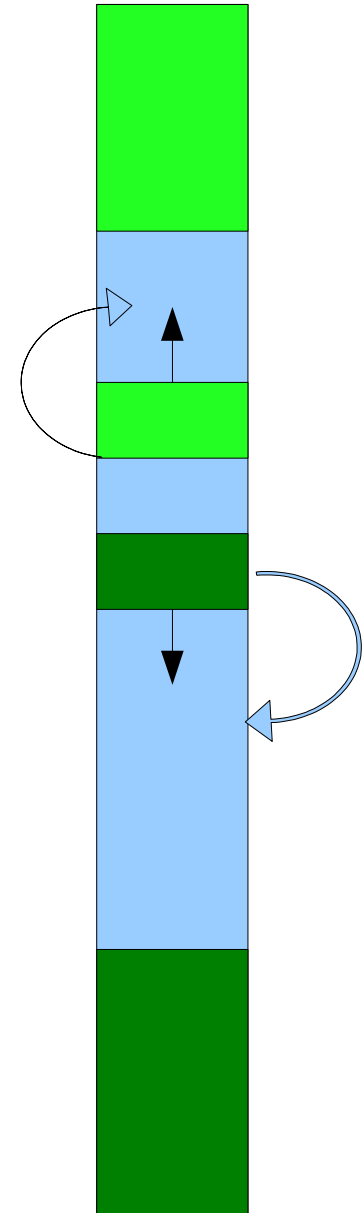
Как это использовать?

# Еще усовершенствование

Пусть  $\min$  — левая граница неотсортированной части массива  
(в начале 0)

Пусть  $\max$  — правая граница неотсортированной части массива  
(в начале N)

- 1) Двигаемся от  $\min$  до  $\max$ . Если встречаем нарушение порядка, меняем местами соседние элементы.
- 2) Сдвигаем  $\max$  на место последней перестановки
- 3) Двигаемся от  $\max$  до  $\min$ . Если встречаем нарушение порядка, меняем местами соседние элементы.
- 4) Сдвигаем  $\min$  на место последней перестановки
- 5) повторяем шаги 1-4 пока  $\min$  и  $\max$  не сойдутся



# Блок-схема

Советую нарисовать перед кодированием  
алгоритма

# Контрольные вопросы

Сложность алгоритма?

$\sim N^2$

Какой случай наихудший?

Обратная отсортированность, сложность  $\sim N^2$

Какой случай наилучший?

Отсортированный массив, сложность  $\sim N$