

# АиСД ДЗ-1

Потякин Арсений

---

## TODO:

1. найти и обосновать инвариант цикла сортировки пузырьком; проанализировать возможности его улучшения
  2. оценить сложность сортировки Шелла для стандартной последовательности интервалов в худшем случае; охарактеризовать конфигурацию массива для худшего случая
- 

## Task 1: (сортировка пузырьком)

- 1) За каждую итерацию внешнего цикла один элемент гарантированно "всплывает" в конец массива, и становится наименьшим среди тех элементов, которые "всплыли" ранее, т.е. становится отсортированным. В данном случае инвариант - это тот самый элемент, и это остается верным на каждой итерации внешнего цикла.
- 2) В случае, если массив уже отсортирован, нужно прекратить выполнение алгоритма. Это можно сделать путем добавления флага: если за одну итерацию вложенного цикла не произошло ни одного "обмена" значениями, то массив отсортирован и можно прервать внешний цикл. Вот реализация улучшенного алгоритма сортировки пузырьком средствами языка Swift:

```
# Swift
func bubbleSort(_ array: inout [Int]) {
    let n = array.count
    var swapped: Bool

    for i in 0..
```

```

    }
    if !swapped {
        break
    }
}
}

```

---

### Task 2: (сортировка Шелла)

1) Стандартная последовательность интервалов -  $\frac{n}{2}, \frac{n}{4}, \dots, 1$

Худший случай - это когда массив отсортирован по убыванию, т.е. обратно отсортирован. Вычислим временную сложность такого массива со стандартными интервалами:

```

// C++
void shellSort(int arr[], int n) {
    // (1)
    for (int gap = n / 2; gap > 0; gap /= 2) {
        // (2)
        for (int i = gap; i < n; i++) {
            int temp = arr[i];
            int j;
            // (3)
            for (j = i; j >= gap && arr[j - gap] > temp; j -= gap) {
                arr[j] = arr[j - gap];
            }
            arr[j] = temp;
        }
    }
}

```

цикл (1): на каждой итерации сортировки значение interval уменьшается в два раза, начиная с  $n/2$ . Это означает, что необходимо совершить

$\log_2 n = O(\log n)$  операций