

N A4

- (1) Ищем элемент, который встречается более чем $(n/2)$ раз (далее будет называться его majority). Т.к. алгоритм использует два вложенных цикла $O(n^2)$, временная сложность $O(n^2)$
 - (2) Также ищем majority элемент. Алгоритм использует только переменные и один цикл от 0 до n , значит временная сложность $O(n)$
 - (3) Также ищем majority элемент. Изначально массив сортируется, а затем проходимся циклом от 0 до n . Значит временная сложность зависит от сортировки. В среднем встроенные сортировки работают за $O(n \log n)$, значит и нашего алгоритма сложность такая.
- Каждый алгоритм ищет majority элемент. Но если такого элемента нет, то результаты выполнения могут отличаться.
 - Например, при $A = [1, 1, 2, 2, 3]$
Алгоритм 1: вернет 1
Алгоритм 2: вернет 3
Алгоритм 3: Условие для возврата не выполнится и цикла не вернется из функции.
 - Как вывод я не согласен с утверждением
 - Алгоритм 1: После подсчета эл-ов добавите проверку на наличие элемента, который является majority
Алгоритм 2: Добавите второй проход по массиву (после первого) для проверки кол-во вхождений
Алгоритм 3: Проверить кол-во вхождений последнего элемента после завершения основного цикла.