1.

a) Данные алгоритмы не решают одну и ту же задачу. Результат работы 3 алгоритма может отличаться от результата работы 1 и 2. 1 и 2 алгоритмы ищут элемент массива, встречающийся более чем n/2 раз, 3 алгоритм ищет элемент массива, встречающийся более чем n/2 раз, но при этом этот элемент не должен быть максимальным в массиве. При этом если в массиве нет элемента, встречающегося больше чем n/2 раз 1 и 3 алгоритмы, ничего не возвращают, а 2 возвращает случайное число.

b)

**A = {0, 1, 1}**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Algorithm1 | | | | |  | Algorithm2 | | |  | Algorithm3 | |
| i | j | c | c1 | ind |  | i | c | ind | i | c |
| 0 | - | 0 | 0 | -1 |  | - | 1 | 0 | - | 1 |
| 0 | 0 | 0 | 1 | -1 |  | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | -1 |  | 2 | 2 | 1 | 2 | 2 |
| 0 | 2 | 0 | 1 | -1 |  |  | | |  | |
| 1 | - | 1 | 0 | 0 |  |
| 1 | 0 | 1 | 0 | 0 |  |
| 1 | 1 | 1 | 1 | 0 |  |
| 1 | 2 | 1 | 2 | 0 |  |
| 2 | - | 2 | 0 | 1 |  |
| 2 | 0 | 2 | 0 | 1 |  |
| 2 | 1 | 2 | 1 | 1 |  |
| 2 | 2 | 2 | 2 | 1 |  |
| - | - | 2 | - | 1 |  |
| return: A[1] = 1 | | | | |  | return: A[1] = 1 | | | return: null | |

**A = {0, 0, 1}**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Algorithm1 | | | | |  | Algorithm2 | | |  | Algorithm3 | |
| i | j | C | c1 | ind |  | i | c | ind | i | c |
| 0 | - | 0 | 0 | -1 |  | - | 1 | 0 | - | 1 |
| 0 | 0 | 0 | 1 | -1 |  | 1 | 2 | 0 | 1 | 2 |
| 0 | 1 | 0 | 2 | -1 |  | 2 | 1 | 0 | 2 | 2 |
| 0 | 2 | 0 | 2 | -1 |  |  | | |  | |
| 1 | - | 2 | 0 | 0 |  |
| 1 | 0 | 2 | 0 | 0 |  |
| 1 | 1 | 2 | 1 | 0 |  |
| 1 | 2 | 2 | 2 | 0 |  |
| 2 | - | 2 | 0 | 0 |  |
| 2 | 0 | 2 | 0 | 0 |  |
| 2 | 1 | 2 | 0 | 0 |  |
| 2 | 2 | 2 | 1 | 0 |  |
| - | - | 2 | - | 0 |  |
| return: A[0] = 0 | | | | |  | return: A[0] = 0 | | | return: A[1] = 0 | |

**A = {0, 1, 2}**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Algorithm1 | | | | |  | Algorithm2 | | |  | Algorithm3 | |
| i | j | c | c1 | ind |  | i | c | ind | i | c |
| 0 | - | 0 | 0 | -1 |  | - | 1 | 0 | - | 1 |
| 0 | 0 | 0 | 1 | -1 |  | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | -1 |  | 2 | 1 | 2 | 2 | 1 |
| 0 | 2 | 0 | 1 | -1 |  |  | | |  | |
| 1 | - | 1 | 0 | 0 |  |
| 1 | 0 | 1 | 0 | 0 |  |
| 1 | 1 | 1 | 1 | 0 |  |
| 1 | 2 | 1 | 1 | 0 |  |
| 2 | - | 1 | 0 | 0 |  |
| 2 | 0 | 1 | 0 | 0 |  |
| 2 | 1 | 1 | 0 | 0 |  |
| 2 | 2 | 1 | 1 | 0 |  |
| - | - | 1 | - | 0 |  |
| return: null | | | | |  | return: A[2] = 2 | | | return: null | |

2.

Algorithm1:

В алгоритме есть два цикла, и один вложен в другой, оба выполняются n раз и содержат константное количество операций, так как все остальные операции являются простыми сравнениями, сложениями и присваиваниями, следовательно, верхней границей можно считать O(n^2).

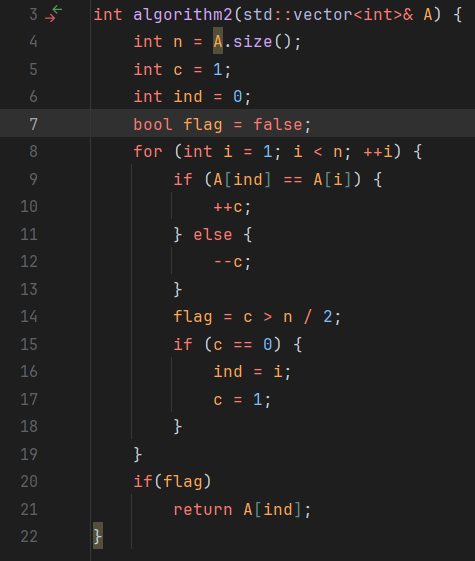
Algorithm2:

В алгоритме один цикл, выполняющийся n раз и содержащий константное количество операций, так как все о

стальные операции являются простыми сравнениями, сложениями и присваиваниями, следовательно, верхней границей можно считать O(n).

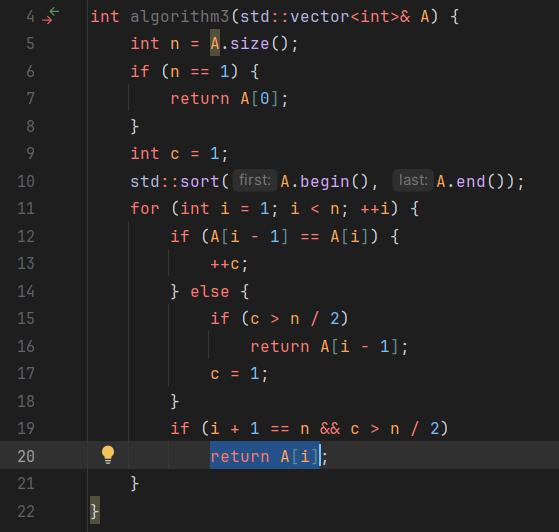
Algorithm3:

В алгоритме один цикл, выполняющийся n раз и содержащий константное количество операций, но при этом так же в алгоритме присутствует сортировка и так как любая сортировка в худшем случае работает не менее чем за O(n), следовательно, верхней границей можно считать O(sort(A)).

3.

Algorithm2:

Необходимо добавить булевою переменную flag с исходным значением false после 2 строки, после 7 строки присваивать переменной значение “c > n /2”. И команду на 12 строке поместить в if с условием flag.

Algorithm3:

Необходимо на 12 строке внутри цикла добавить if с условием “ i + 1 = n && c > n / 2” внутри которого будет команда “ return A[i]”

4.

Algorithm2:

Добавление новой перменной добавляет +1 операцию что никак не повлияет на линейную верхнюю границу, аналогично и добавление if. Присваивание и вычисление значения переменной добавляет +3n операций, что никак не сказывается на линейной верхней границе алгоритма, следовательно верхней границей можно считать O(n).

Algorithm3:

Добавление if добавляет (n + 1) операцию что никак не сказывается на линейной верхней границе и так как любая сортировка в худшем случае работает не менее чем за O(n), следовательно, верхней границей можно считать O(sort(A)).