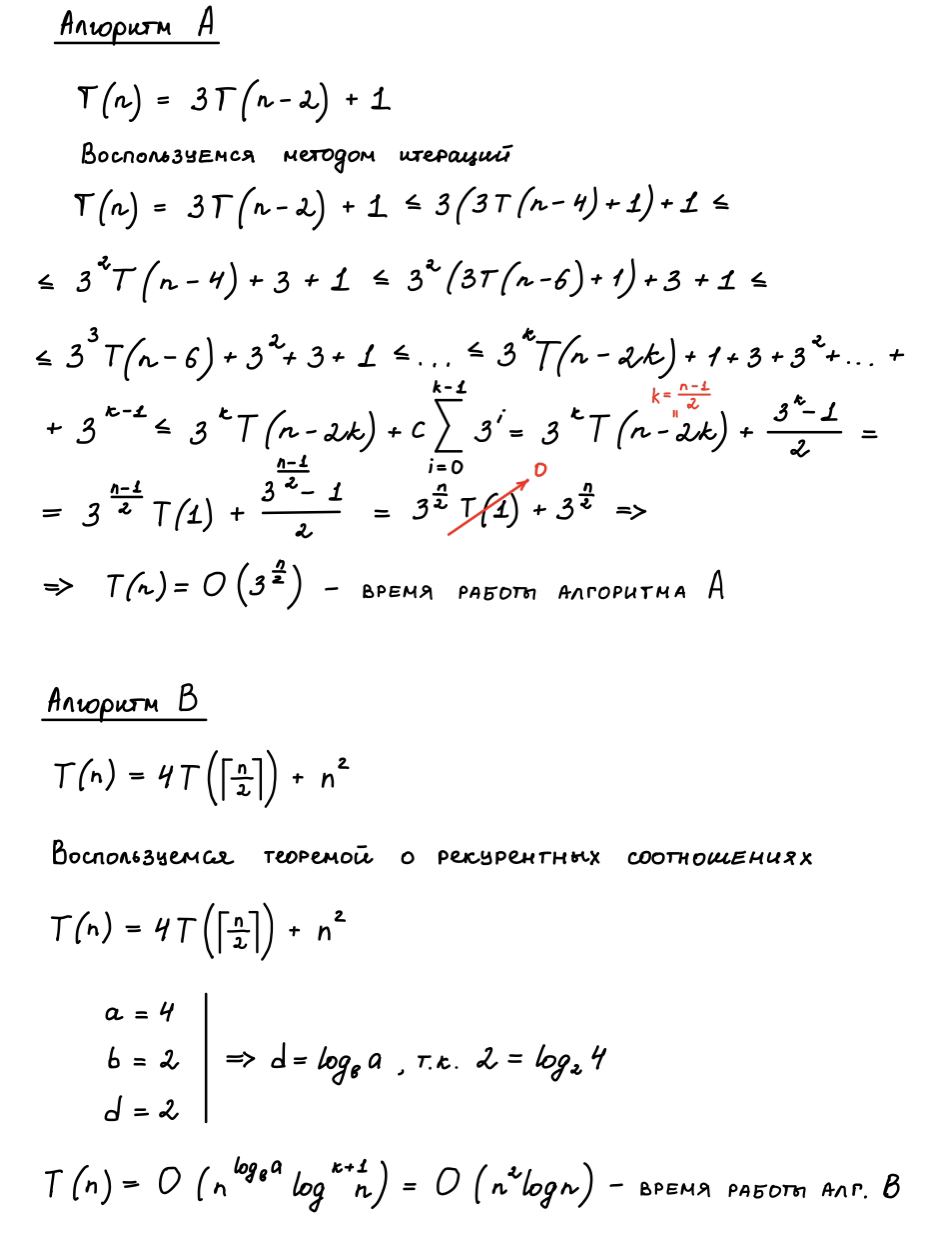
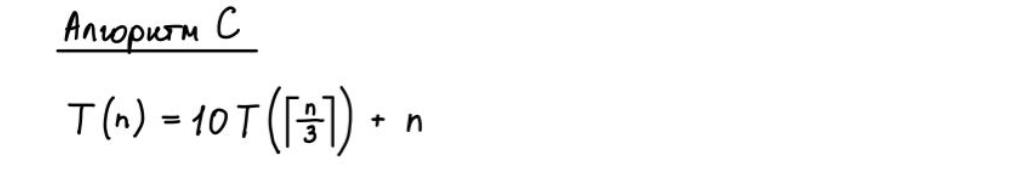
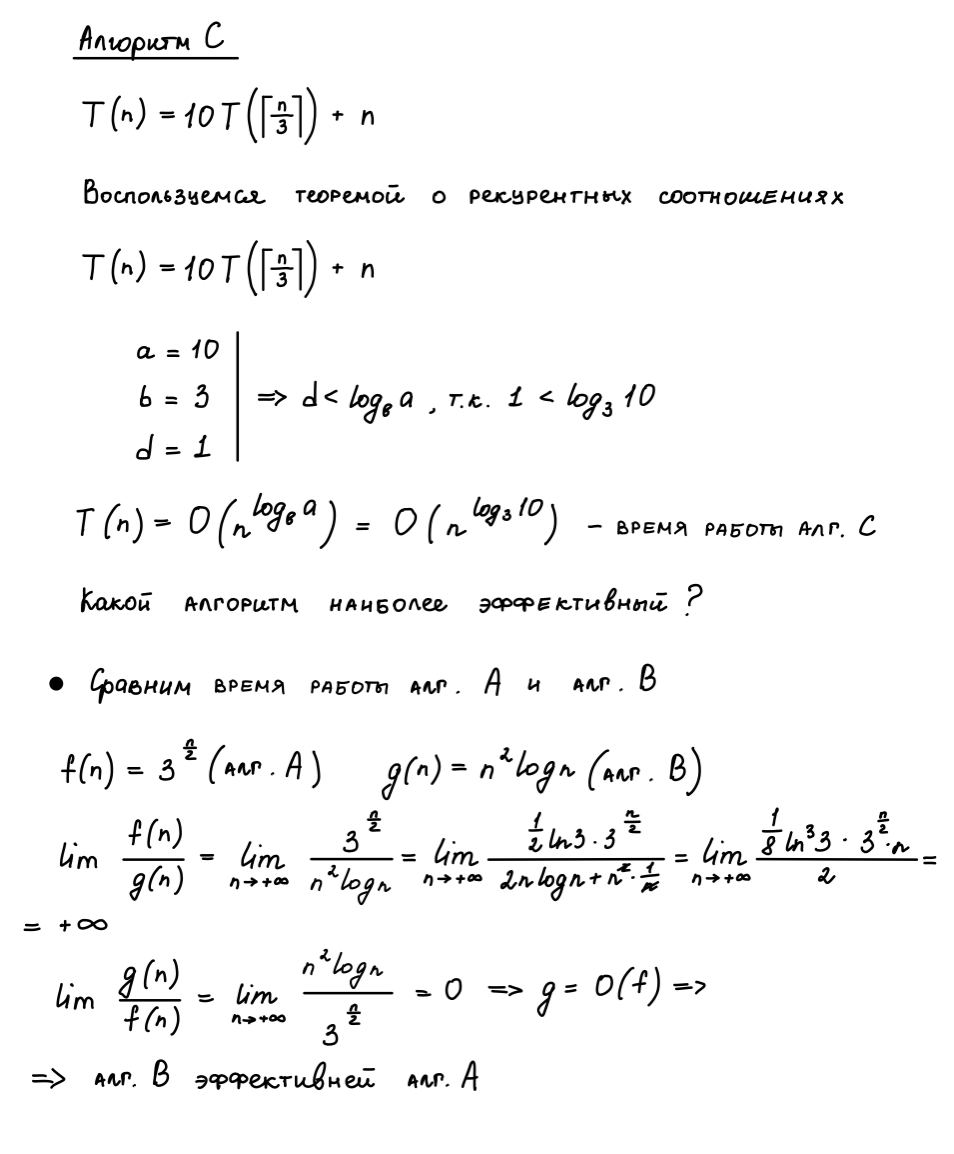
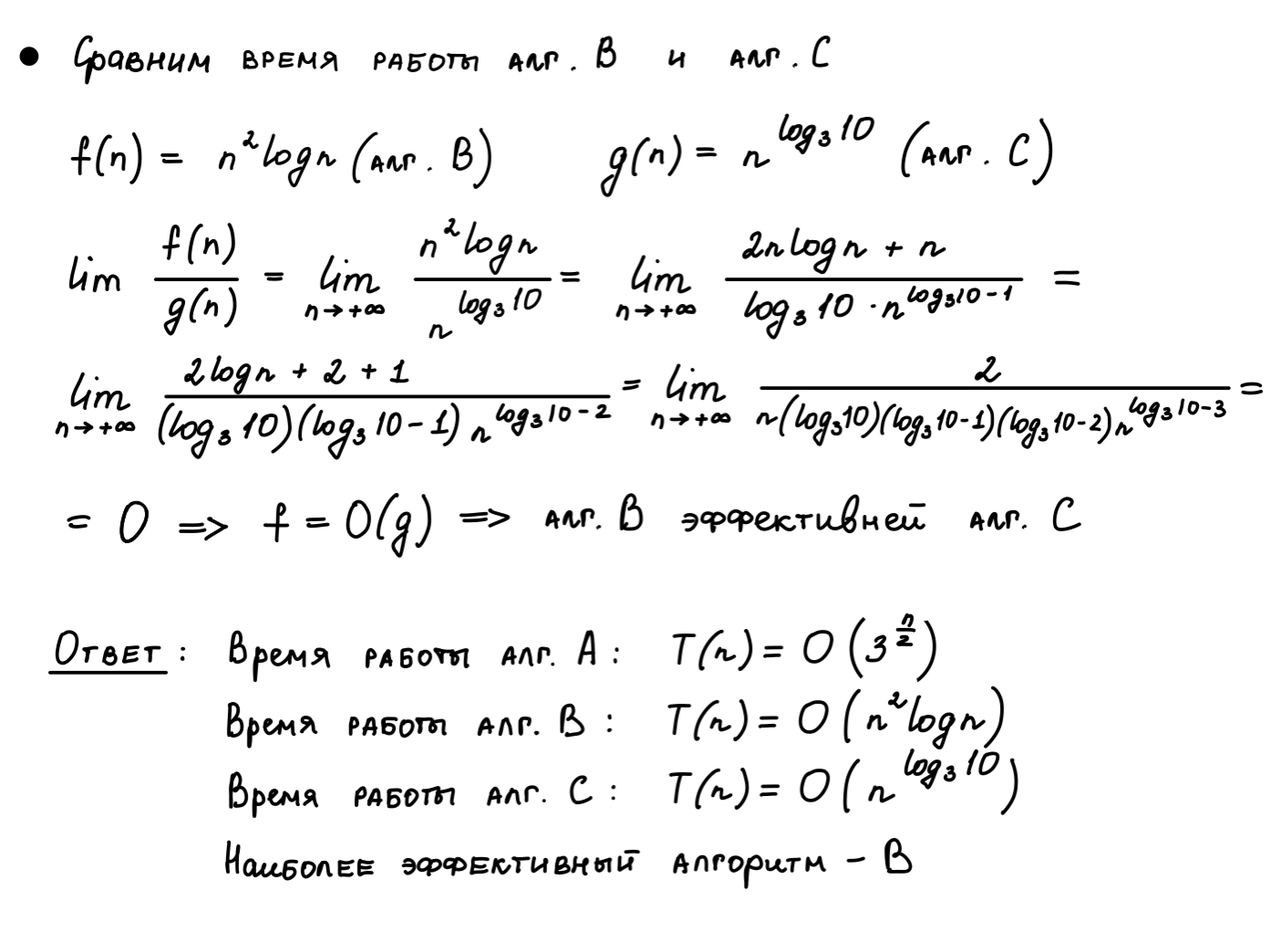
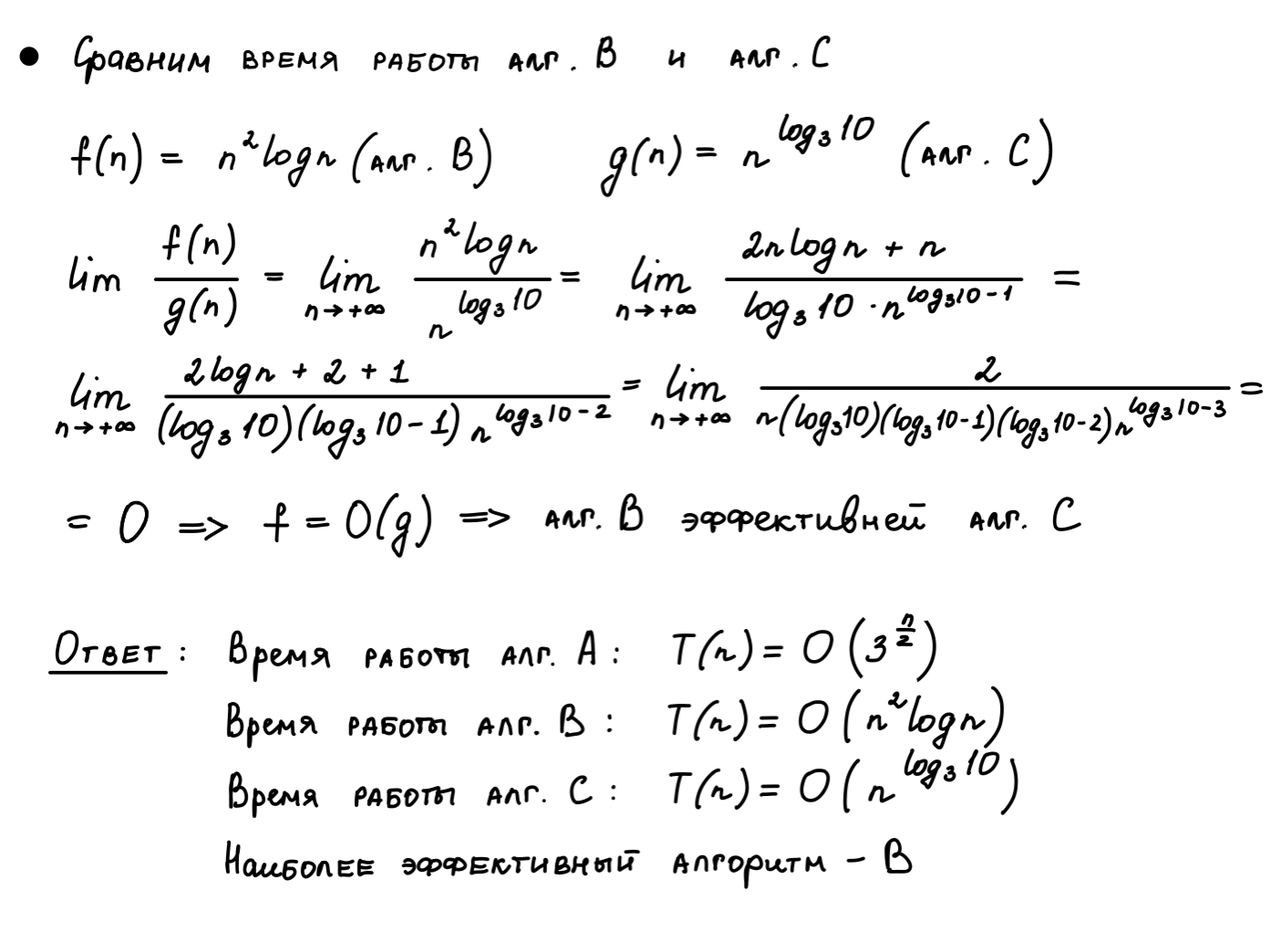
Задача 1



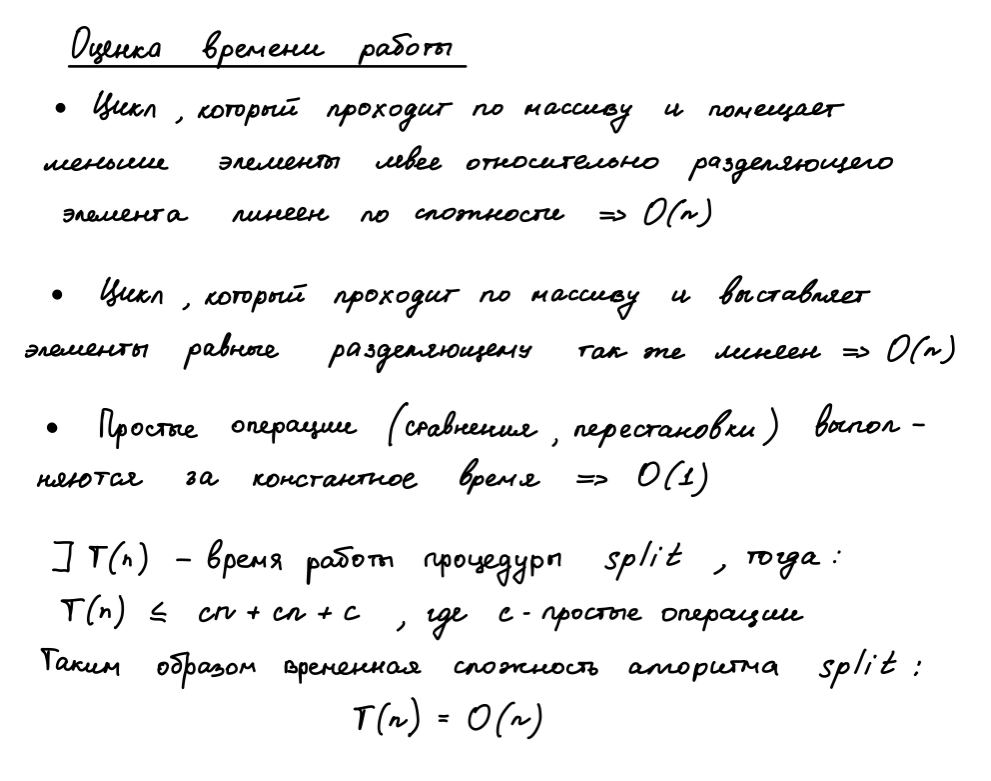
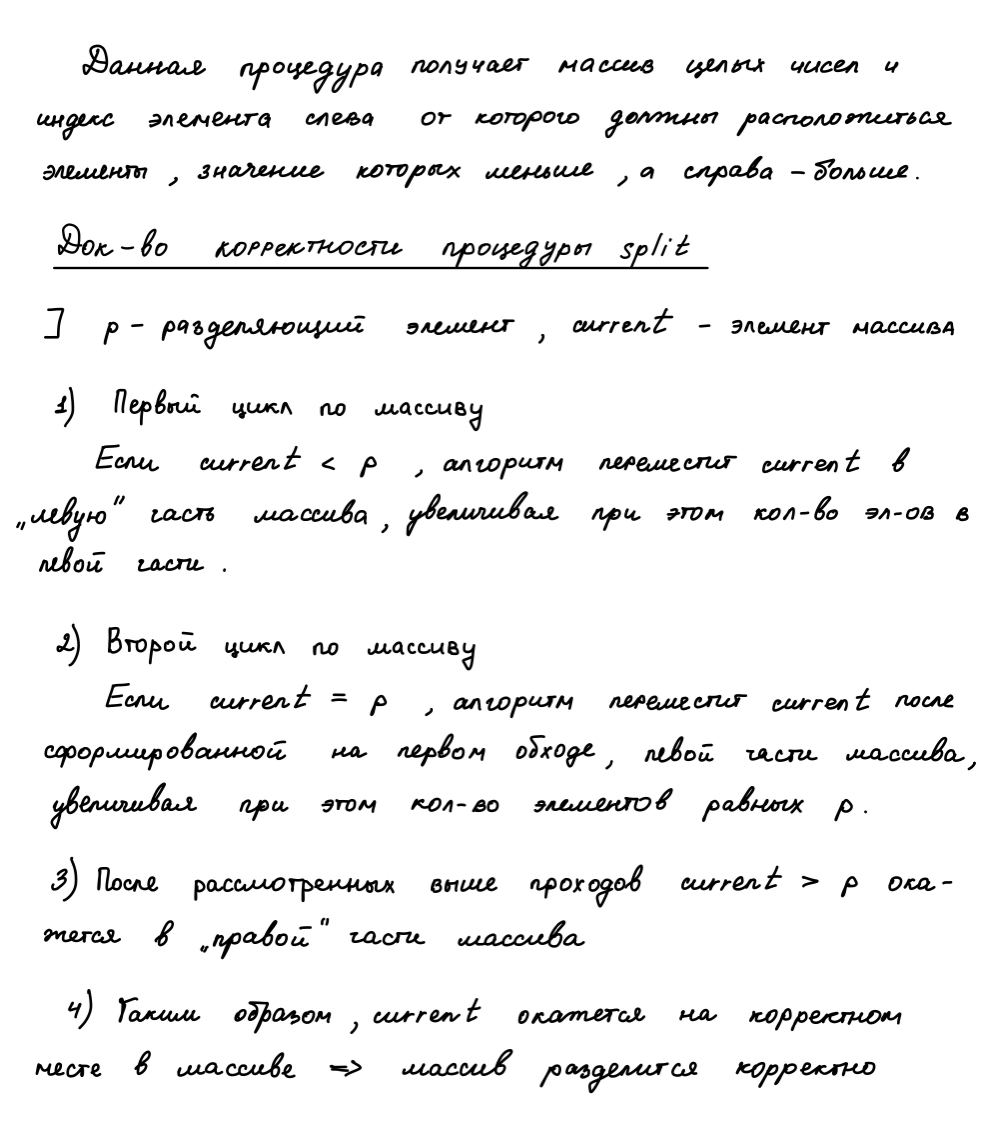


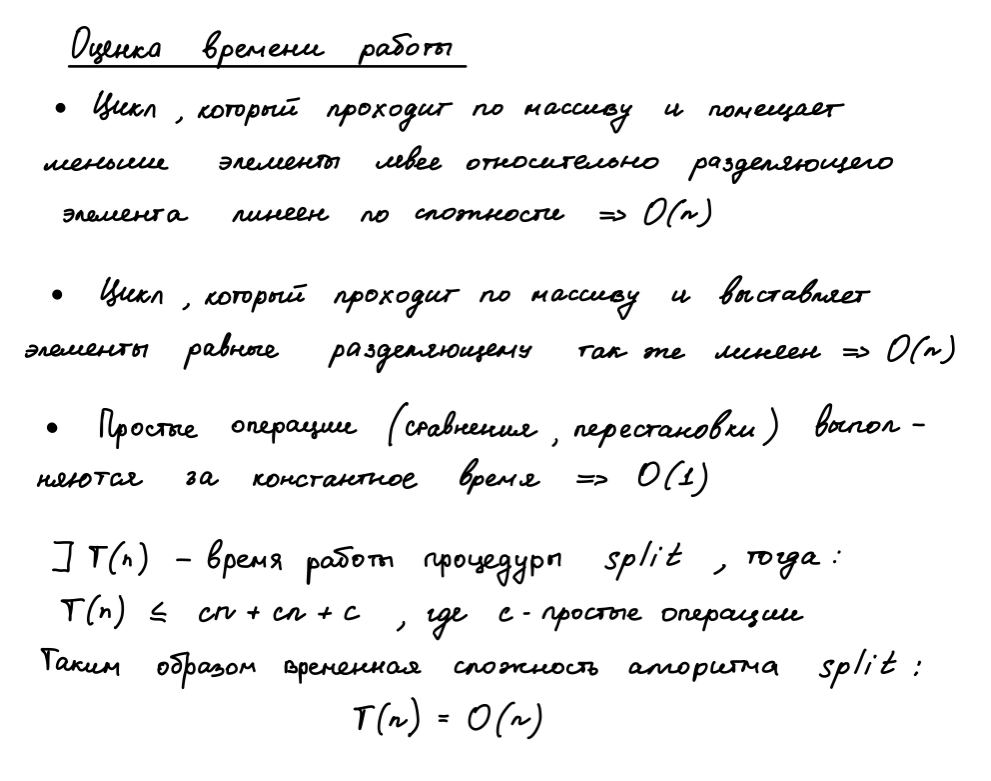






Задание 3





#include <iostream>

#include <vector>

#include <cmath>

**void** printArray(std::vector<**int**> mass)

{

std::cout << '\n';

**for** (**unsigned** **int** i = 0; i < mass.size(); i++)

{

std::cout << mass[i] << " ";

}

std::cout << '\n';

}

std::pair<**int**, **int**> split(std::vector<**int**> &array, **int** index)

{

**int** p = array[index];

**int** elemLessProp = 0;

**int** elemEqualProp = 0;

**for** (**int** i = 0; i < array.size(); i++)

{

**if** (array[i] < p)

{

**if** (elemLessProp == i)

{

elemLessProp++;

}

**else**

{

std::swap(array[i], array[elemLessProp]);

elemLessProp++;

}

}

}

elemEqualProp = elemLessProp;

**for** (**int** i = elemEqualProp; i < array.size(); i++)

{

**if** (array[i] == p)

{

**if** (elemEqualProp == i)

{

elemEqualProp++;

}

**else**

{

std::swap(array[i], array[elemEqualProp]);

elemEqualProp++;

}

}

}

**return** std::make\_pair(elemLessProp, elemEqualProp);

}

**int** median(std::vector<**int**> array, **int** k)

{

**if** (array.size() == 1)

{

**return** array[0];

}

**int** randomElement = rand() % (array.size());

std::pair<**int**, **int**> prop = split(array, randomElement);

**int** elemLessProp = prop.first;

**int** elemEquelProp = prop.second;

**if** (k <= elemLessProp)

{

std::vector<**int**> arrayLeft;

arrayLeft.insert(arrayLeft.end(), array.begin(), array.begin() + elemLessProp);

**return** median(arrayLeft, k);

}

**else** **if** (k > elemLessProp + (elemEquelProp - elemLessProp))

{

std::vector<**int**> arrayRight;

arrayRight.insert(arrayRight.end(), array.begin() + elemEquelProp, array.end());

**return** median(arrayRight, k - elemLessProp - (elemEquelProp - elemLessProp));

}

**return** array[randomElement];

}

**int** main(**int** argc, **char**\*\* argv)

{

size\_t size = 0;

std::vector<**int**> array;

**int** elem = 0;

**int** randomIndex = 0;

std::cin >> size;

**for** (**int** i = 0; i < size; i++)

{

std::cin >> elem;

array.push\_back(elem);

}

std::cin >> randomIndex;

**int** indexOfMedian = **int**(array.size() / 2);

**if** (array.size() % 2 != 0)

{

++indexOfMedian;

}

std::pair<**int**, **int**> indexes = split(array, randomIndex);

printArray(array);

std::cout << indexes.first << " " << indexes.second - 1;

std::cout << "\nmedian: " << median(array, indexOfMedian);

std::cout << "\n\n";

**return** 0;

}

Тесты программы из задачи 3

|  |  |
| --- | --- |
| Input | Output |
| 1  1  0 | 1  0 0  median: 1 |
| 2  2 1  0 | 1 2  1 1  median: 1 |
| 3  4 2 5  2 | 4 2 5  2 2  median: 4 |
| 4  8 7 6 5  2 | 5 6 7 8  1 1  median: 6 |
| 5  1 1 6 5 4  2 | 1 1 5 4 6  4 4  median: 4 |
| 6  2 2 1 3 2 4  1 | 1 2 2 2 3 4  1 3  median: 2 |
| 7  0 1 2 3 6 5 4  2 | 0 1 2 3 6 5 4  2 2  median: 3 |
| 8  4 4 3 5 5 5 6 6  3 | 4 4 3 5 5 5 6 6  3 5  median: 5 |
| 9  4 2 5 7 8 9 1 6 3  6 | 1 2 5 7 8 9 4 6 3  0 0  median: 5 |
| 10  1 2 1 2 1 2 1 2 1 2  6 | 1 1 1 1 1 2 2 2 2 2  0 4  median: 1 |