**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЯ ОБРАЗОВАНИЯ**

**“ПОЛОЦКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ”**

Факультет информационных технологий

Кафедра технологий программирования

**Лабораторная работа №6**

Выполнила студент 2 курса, группа 21-ИТ-1 Катушёнок И.В.

Проверила Виноградова А.Д

Полоцк, 2022 г.

**Задание 1:** В первой строке даны целое число 1 ≤ n ≤ 105 и массив A[1…n] из n различных натуральных чисел, не превышающих 109, в порядке возрастания, во второй — целое число 1 ≤ k ≤ 105 и k натуральных чисел b1,…,bk не превышающих 109. Для каждого i от 1 до k необходимо вывести индекс 1 ≤ j ≤ n, для которого A[j]=bi​, или −1, если такого j нет.

#include <algorithm>

#include <cassert>

#include <iostream>

#include <vector>

int get\_pos(const std::vector<int>& numbers, int number) {

int l = 0;

int r = numbers.size() - 1;

while (l <= r) {

int m = (l + r) / 2;

if (numbers[m] == number)

return m + 1;

else if (numbers[m] > number)

r = m - 1;

else l = m + 1;

}

return -1;

}

int main(void) {

size\_t number\_count;

std::cin >> number\_count;

std::vector<int> numbers(number\_count);

for (auto& number : numbers) {

std::cin >> number;

}

assert(std::is\_sorted(numbers.begin(), numbers.end()));

size\_t query\_count;

std::cin >> query\_count;

while (query\_count-- > 0) {

int number;

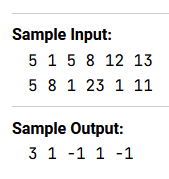
std::cin >> number;

std::cout << get\_pos(numbers, number) << " ";

}

std::cout << std::endl;

}

****

**Задание 2:** Первая строка содержит число 1 ≤ n ≤ 105, вторая — массив A[1…n], содержащий натуральные числа, не превосходящие 109. Необходимо посчитать число пар индексов 1 ≤ i < j ≤ n, для которых A[i] > A[j]. (Такая пара элементов называется инверсией массива. Количество инверсий в массиве является в некотором смысле его мерой неупорядоченности: например, в упорядоченном по неубыванию массиве инверсий нет вообще, а в массиве, упорядоченном по убыванию, инверсию образуют каждые два элемента.)

#include <vector>

#include <iomanip>

#include <iostream>

#include <ios>

using namespace std;

vector <long> merge(vector <long> a, vector<long> b, long& l) {

long inda = 0, indb = 0;

long sa = a.size(), sb = b.size();

vector <long> c(sa + sb);

for (long i = 0; i < sa + sb; i++) {

if (inda == sa) {

c[i] = b[indb];

indb++;

}

else if (indb == sb) {

c[i] = a[inda];

inda++;

}

else if (a[inda] <= b[indb]) {

c[i] = a[inda];

inda++;

}

else {

c[i] = b[indb];

indb++;

l += sa - inda;

}

}

return c;

}

int main()

{

long k, kol = 0;

cin >> k;

vector<vector<long>> Q(k);

vector <long> a(1);

for (long i = 0; i < k; i++) {

cin >> a[0];

Q[i] = a;

}

long t = 0;

long S = Q.size();

while (S > 1) {

Q[t] = merge(Q[t], Q[t + 1], kol);

t += 2;

if (t >= S - 1) {

t = 0;

long y = 0;

while (2 \* y < S) {

Q[y] = Q[2 \* y];

y++;

}

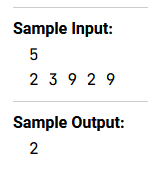
S = (S + 1) / 2;

}

}

cout << kol;

}



**Задание 3:** В первой строке задано два целых числа 1 ≤ n ≤ 50000 и 1 ≤ m ≤ 50000 — количество отрезков и точек на прямой, соответственно. Следующие n строк содержат по два целых числа ai​ и bi​ (ai ≤ bi​) — координаты концов отрезков. Последняя строка содержит m целых чисел — координаты точек. Все координаты не превышают 108 по модулю. Точка считается принадлежащей отрезку, если она находится внутри него или на границе. Для каждой точки в порядке появления во вводе выведите, скольким отрезкам она принадлежит.

#include <iostream>

#include <algorithm>

#include <vector>

using namespace std;

int get\_pos(const std::vector<long long>& numbers, int number, bool mode\_l = true) {

int l = 0;

int r = numbers.size() - 1;

while (l <= r) {

int m = (l + r) / 2;

if (numbers[m] > number && mode\_l)

return m;

else if (numbers[m] > number && mode\_l == false)

return m;

else

l = m + 1;

}

return (l - 1);

}

int main() {

int n, m;

cin >> n >> m;

vector<long long> lefts(n);

vector<long long> rights(n);

for (int i = 0; i < n; i++)

cin >> lefts[i] >> rights[i];

sort(lefts.begin(), lefts.end());

sort(rights.begin(), rights.end());

for (int i = 0; i < m; i++) {

int num, k1 = 0, k2 = 0;

cin >> num;

k1 = get\_pos(lefts, num);

while (k1 >= 0 && lefts[k1] > num)

k1--;

k2 = get\_pos(rights, num, false);

while (k2 >= 0 && rights[k2] >= num)

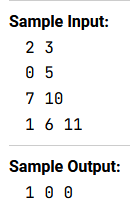
k2--;

cout << (k1 - k2) << ' ';

}

return 0;

}



**Задание 4:** Первая строка содержит число 1 ≤ n ≤ 104, вторая — n натуральных чисел, не превышающих 10. Выведите упорядоченную по неубыванию последовательность этих чисел.

#include <iostream>

#include <vector>

std::vector<int> count\_sort(std::vector<int>& A) {

const int MAX\_NUM = 10000;

int B[MAX\_NUM + 1] = { 0 };

std::vector<int> result(A.size());

// Count freq

for (size\_t i = 0; i < A.size(); ++i) { ++B[A[i] - 1]; }

// Cumulative sum

for (size\_t i = 1; i <= MAX\_NUM; ++i) { B[i] += B[i - 1]; }

for (int i = A.size() - 1; i >= 0; --i) {

result[B[A[i] - 1] - 1] = A[i];

--B[A[i] - 1];

}

return result;

}

int main() {

int n;

std::cin >> n;

std::vector<int> A(n);

for (int i = 0; i < n; ++i) { std::cin >> A[i]; }

A = count\_sort(A);

for (int i = 0; i < n; ++i) { std::cout << A[i] << " "; }

return 0;

}

