ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«Национальный исследовательский университет ИТМО»**

**Факультет программной инженерии и компьютерной техники**

«Алгоритмы и структуры данных»

**отчет по блоку задач №2 (Яндекс.Контест)**

**Выполнил:**

Студентка группы P3215

Шмидт А.А.

**Преподаватель:**

Тараканов Д.С

Санкт-Петербург, 2025г.

**Задачи из 2-го блока Яндекс.Контеста**

**Задача №1 «E. Коровы в стойла»**

Описание:

Требуется найти наименьшее максимальное расстояние, на котором можно поставить коров в стойла.

Основная сложность заключается в том, чтобы догадаться, что в задаче нужно использовать бинарный поиск, дальше она +- интуитивно решается: просто делается пробег по вектору с поиском подходящих под условие элементов.

Алгоритмическая сложность:

1. **По времени:** есть бинарный поиск по ответу(O(log(n)) и линейный проход по массиву(O(n)), так что суммарные затраты: O(nlog(n))
2. **По памяти:** память занимается вектором O(n) и переменными O(1), так что O(n)

Код:

#include <iostream>

#include <vector>

int main() {

int n = 0;

int k = 0;

std::cin >> n >> k;

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

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

std::cin >> arr[i];

}

if (k == 2) {

std::cout << arr[n - 1] - arr[0];

return 0;

}

int min = 0;

int max = arr[n - 1] - arr[0] + 1;

while (max - min > 1) {

int med = (min + max) / 2; *// бин поиск*

int done = 1;

int last = arr[0];

for (int i = 1; i < n; i++) {

if (arr[i] - last >= med) {

done++;

last = arr[i];

}

}

if (done >= k) {

min = med;

} else {

max = med;

}

}

std::cout << min;

}

}

**Задача №2 «F. Число»**

Описание: необходимо найти максимальное число, которое может получиться при перестановке входных чисел. Логика в том, чтобы рассматривать вектор входных данных именно в строковом формате, потому что так они эффективнее сортируются и потом проводить дополнительную сортировку для определения «спорных» ситуаций.

Алгоритмическая сложность:

1. **По времени:** добавление данных в вектор О(n), сортировка работает за O(n\*log(n)), двойной вложенный цикл – O(n\*\*2), итоговая сложность O(nlog(n) + n\*\*2)
2. **По памяти:** вектор хранит n строк, так что O(n), переменные занимают O(1), так что O(n)

**Код:**

#include <algorithm>

#include <iostream>

#include <string>

#include <vector>

int main() {

std::vector<std::string> arr;

std::string input;

std::string buffer;

while (std::cin >> buffer) {

arr.push\_back(buffer);

}

std::sort(arr.begin(), arr.end());

for (size\_t i = 0; i < arr.size(); i++) {

for (size\_t j = i + 1; j < arr.size(); j++) {

if (arr[i] + arr[j] > arr[j] + arr[i]) {

std::swap(arr[i], arr[j]);

}

}

}

for (size\_t i = arr.size(); i > 0; i--) {

std::cout << arr[i - 1];

}

}

**Задача №4 «H. Магазин»**

Описание: необходимо понять, какую наименьшую сумму человек потратит при совершении покупок, если каждый k-ый товар – бесплатный. На самом деле задача даже проще, чем ЕГЭ-шная 26-я, так что решение прямо в лоб: сортировка и чеки, в которых мы будем покупать ближайшие по стоимости товары.

Алгоритмическая сложность:

1. **По времени:** чтение O(n), сортировка O(nlog(n)), проход по массиву O(n) => O(nlog(n))
2. **По памяти:** вектор занимает O(n), переменные O(1) => O(n)

Код:

#include <algorithm>

#include <iostream>

#include <vector>

int main() {

int n = 0;

int k = 0;

int sum = 0;

std::cin >> n >> k;

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

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

std::cin >> arr[i];

}

std::sort(arr.rbegin(), arr.rend());

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

if ((i + 1) % k != 0) {

sum += arr[i];

}

}

std::cout << sum;

}