Университет ИТМО

Факультет программной инженерии и компьютерной техники

**Лабораторная работа №2**

по «Алгоритмам и структурам данных»

Timus

Выполнил:

Студент группы P3233

Фамилия И.О.

Шикунов Максим Евгеньевич

Преподаватели:

Косяков М.С.

Тараканов Д.С.

Санкт-Петербург

2024

Задача №1322 «Шпион»

Код:

#include <iostream>  
#include <set>  
using namespace std**;**int main() {  
 int n**;** string s**;** cin >> n >> s**;** multiset<pair<char**,** int>> chars**;** int index = 0**,** i = 0**,** count = 0**;** for (char ch : s) {  
 chars.insert(pair(ch**,** index))**;** index++**;** count++**;** }  
 pair<char**,** long> arr[count]**;** for (pair<char**,** long> p : chars) {  
 arr[i] = p**;** i++**;** }  
 index = n - 1**;** for (int j = 0**;** j < s.size()**;** j++) {  
 cout << arr[index].first**;** index = arr[index].second**;** }  
 cout << endl**;** return 0**;**}

Пояснения к примененному алгоритму:

Принимаем числа и записываем их в пары с индексом, в порядке котором они были поданы на вход. Дальше сортируем их по порядку, в котором они находятся в первой позиции (то есть у наших слов первые буквы будут все в порядке возрастания). Первым число выводим букву по индексу, который нам дали на вход, а дальше у этого числа берем изначальный индекс (то есть номер слова в котором эта буква находится в конце слова) и выводим первую букву этого слова. Повторяем это пока не выведем строку нашей длины.

Сложность:

Задача №1726 «Кто ходит в гости…»

Код:

#include <iostream>  
#include <set>  
using namespace std**;**int main() {  
 size\_t n**,** sumi = 0**,** count**;** cin >> n**;** count = n \* (n - 1)**;** multiset<long> coordX**,** coordY**;** long arrX[n]**,** arrY[n]**;** long x**,** y**;** for (size\_t i = 0**;** i < n**;** i++) {  
 cin >> x >> y**;** coordX.insert(x)**;** coordY.insert(y)**;** }  
 long countNum = 0**;** for (long i : coordX) {  
 arrX[countNum] = i**;** countNum++**;** }  
 countNum = 0**;** for (long i : coordY) {  
 arrY[countNum] = i**;** countNum++**;** }  
 for (long i = n - 1**;** i > 0**;** i--) {  
 sumi += (arrX[i] - arrX[i - 1] + arrY[i] - arrY[i - 1]) \* (n - i) \* i \* 2**;** }  
 cout << sumi / count**;** return 0**;**}

Пояснение к алгоритму:

В данной задаче мы сортируем наши координаты X и Y для того, чтобы переделать координаты входных точке так, чтобы они лежали в одном направлении. Это нам поможет, потому что теперь чтобы дойти от первой точки до третей, мы точно будет проходить через вторую и так далее (если будем идти от 1 до n, то пройдем 2, 3, …, n-1 точки). Тем самым, зная сколько у нас точек, мы можем посчитать, сколько каждый промежуток студенты буду проходить и тем самым посчитать окончательную сумму.

Сложность: