

Университет ИТМО

Факультет программной инженерии и компьютерной техники

Лабораторная работа №1

по «Алгоритмам и структурам данных»

Базовые задачи / Timus

Выполнил:

Студент группы Р3232

Чмурова М. В.

Преподаватели:

Косяков М.С.

Тараканов Д.С.

Санкт-Петербург

2024

Задача №А «Агроном-любитель»

Пояснение к примененному алгоритму:

Необходимо пройти по массиву значений и найти максимальную длину, где нет трех идущих подряд повторяющихся символов.

Для этого начинаем идти по массиву со второго элемента и проверять – не имеют ли элементы справа и слева от него то же значение (поэтому отдельно проверяем не являются ли текущий элемент последним в массиве, чтобы не выйти за границу массива).

При выполнении этого условия сохраняем индекс последнего элемента (текущего) и проверяем больше ли полученный отрезок максимального (длину максимального изначально задаем равную нулю). Сохраняем новую максимальную длину и значения индекса первого и последнего элемента, если условие выполняется, и продолжаем идти дальше по массиву.

После чего выводим полученные максимальную длину и индексы.

Задача №В «Зоопарк Глеба»

Пояснение к примененному алгоритму:

Для того, чтобы определить можно ли загнать всех животных в ловушку, необходимо пройти по массиву букв и найти пары, где одна буква большая, а вторая маленькая и «удалить» их из строки. Эту операцию необходимо повторять, пока вся строка не станет пустой. Если же с помощью такого алгоритма сделать строку пустой невозможно, то выводим «Impossible».

Для реализации этого алгоритма необходимо иметь массив с индексами букв, у которых еще нет пар. Также нужны точно такие же массивы: один для больших букв и один для маленьких. Они будут хранить в себе соответственно номер большой или маленькой буквы, это необходимо для того, чтобы сохранять индекс маленькой буквы по номеру большой буквы.

Задача №С «Конфигурационный файл»

Пояснение к примененному алгоритму:

В третьей задаче необходимо написать парсер, который будет хранить изменения текущего блока и удалять их, если блок заканчивается. Таким образом, необходимо хранить изменения для каждого блока как массив массивов, где хранятся названия переменных, которые претерпели изменения в текущем блоке.

При открытии скобочки (переходе на новый блок) понадобится новый массив для изменений нового блока. А при закрытии скобочки (блок закончился) понадобится удалить все последние значения переменных, которые претерпели изменения.

Задача №D «Профессор хаус»

Пояснение к примененному алгоритму:

В задаче на вычисление количества бактерий все делалось в лоб: совершаем вычисления – количество бактерий для текущего этапа. Если значение получилось меньше 1, то последующие вычисления не имеют смысла – конечное значение на всех последующих этапах будет ноль, что и возвращается.

Также необходимо проверять не является ли вычисленное новое количество бактерий равным количеству предыдущего этапа: в таком случае дальнейшие вычисления также не имеют смысла, поскольку ответ на следующих этапах всегда будет одинаковым. Поэтому возвращаем это значение и завершаем работу.

Задача №1401 «Игроки»

Пояснение к примененному алгоритму:

Необходимо использовать рекурсивный подход: изначальный массив разбивается на 4 квадранта, где для каждого квадранта происходит то же самое, пока не получится матрица 2 на 2 элемента.

Для каждой матрицы есть массив индексов, которые в ней используются – это необходимо для того, чтобы определить четыре центральных элемента матрицы – исходя из них «диагональные» центральные элементы (те, которые не входят в квадрант с выбитой ячейкой) заполняются значением счетчика, после чего для каждого из этих элементов вызывается рекурсивная функция где «выбитой» ячейкой считается тот самый центральный элемент. Для центрального элемента, который содержится в одном квадранте с выбитой ячейкой, вызывается функция, где вместо центрального элемента передается та же выбитая ячейка»

Таким образом, массив заполнится пройдя рекурсивно по все возможным квадрантам, уменьшая их до квадранта 2x2

Задача №2025 «Стенка на стенку»

Пояснение к примененному алгоритму:

Для вычисления количества схваток необходимо добавлять значение схваток для каждого нового бойца, которое вычисляется так: из количества всех бойцов вычитается количество бойцов в текущей команде (так как бойцы из одной команды не могут сражаться друг с другом, а для одного бойца будут ожидать схватки со всеми остальными участниками)