

Задачи

Дан полный взвешенный граф. Найти в нем минимальный гамильтонов путь.

Реализовать симулятор сервиса по продаже билетов в кинотеатре. Кинотеатр это поле $N \times M$. Есть L зон, непересекающихся, каждая со своей ценой. Зона — последовательный набор рядов.

Запросы:

Купить X мест, в линию или прямоугольник, бронь идет на группу с названием C ;

Купить X мест, как можно ближе к экрану, но важно сесть вместе;

Купить одно место;

Снять бронь взятую группой с названием C ;

Напечатать, инициализировать, обнулить зал;

Подсчитать прибыль.

Давайте попробуем посчитать число разбиений числа n в сумму слагаемых, каждое из которых не больше k . Совершенно очевидно, что ответ это рекуррента вида

$ans_n = ans_{n-1} + ans_{n-2} + \dots + ans_{n-k}$. Естественно, решение за k^n , то есть наивная рекурсия, работает слишком медленно.

Задача состоит в том, чтобы посчитать ответ быстрее.

Напишите бота, в худшем случае сводящего к ничьей партию в крестики-нолики 3×3 .

Реализовать в объектах библиотеку по работе с матрицами.

Умножение на матрицу, на вектор;

Печать;

Вычисление следа;

Транспонирование;

Вычисление решения методом гаусса уравнения $A * x = B$,
строки ЛНЗ;

Перегрузить арифметические операторы.

- а) Посчитать значение выражения по его обратной польской записи;
- б) Найти обратную польскую запись по выражению.

Дана формула, в которой между скобками только конъюнкция, между переменными в скобках только дизъюнкция, и в каждой скобке ровно 2 переменных. Необходимо найти значения переменных, при которых формула обращается в истину.

Дана строка s . Вычислить массив P , где $P[i]$ равно длине наибольшего префикса подстроки $s[0..i-1]$, совпадающего с ее суффиксом. При этом всю подстроку в качестве префикса брать нельзя.

Рассмотрим алгоритм Евклида (с делением с остатком, а не с вычитанием) и число n . Алгоритм прекращает свою работу, когда одно из чисел становится равно 0.

а) Найти два числа, для которых алгоритм Евклида делает ровно n шагов.

б) Найти два многочлена с коэффициентами $-1, 0, 1$, для которых алгоритм Евклида делает ровно n шагов.

Реализуйте метод наименьших квадратов.

Дан массив целых чисел, найдите подмножество не связанных между собой элементов с максимальной суммой. Вычислите сумму этого подмножества. Например, задан массив $arr = [-2, 1, 3, -4, 5]$ у нас есть следующие возможные подмножества:

Subset	Sum
$[-2, 3, 5]$	6
$[-2, 3]$	1
$[-2, -4]$	-6
$[-2, 5]$	3
$[1, -4]$	-3
$[1, 5]$	6
$[3, 5]$	8

— максимальная сумма 8.

В бесконечной матрице в клетку $[0,0]$ записывается 0. В клетку $[i,j]$ записывается минимальное число, которое не присутствует в i -ой строке и j -ом столбце. Написать функцию, которая по входящим параметрам $i \geq 0$ и $j \geq 0$ возвращала бы значение ячейки $[i,j]$.

Пример таблицы 5 на 5:

0	1	2	3	4
1	0	3	2	5
2	3	0	1	6
3	2	1	0	7
4	5	6	7	0

Проверить, присутствует ли в строке такая буква, при удалении которой строка станет палиндромом. Если она изначально палиндром или из неё нельзя сделать палиндром, то вывести -1.

Дана матрица n на k , написать программу, приводящую эту матрицу к ступенчатому виду при помощи элементарных преобразований над строками. Элементарные преобразования над строками бывают трех видов: поменять две строки местами, умножить все элементы какой-либо строки на одно и то же ненулевое число, добавить каждый элемент одной строки с одним и тем же коэффициентом к элементу другой строки, стоящему в том же столбце.

За константную память найти в массиве самое редкое число.

- а). Все числа встречаются дважды, одно число один раз;
- б). Все числа встречаются трижды, одно число один или два раза;
- в). Все числа встречаются k раз, одно число l раз,
 $k > l, l > 0$, k и l не известны.

Процессор называется Big Endian, если целые числа, занимающие несколько байтов, хранятся по следующему правилу: младший байт записан по наибольшему адресу, и Little Endian, если младший байт хранится по наименьшему адресу; если же порядок записи байтов в составе целого числа не совпадает ни с одним из упомянутых, назовем это Mixed Endian. Определить тип процессора, на котором мы работаем.

Написать функцию `create`, заводящую блочно-диагональную матрицу заданного порядка (он определяется во время выполнения программы; порядок блоков равен 10; хранить в памяти, естественно, нужно только возможно ненулевые элементы) в соответствии со вторым подходом к двумерным динамическим массивам (возвращает `int**`). Написать также функцию `access`, выдающую ссылку на заданный элемент этой матрицы по указателю на такую матрицу (типа `int**`) и номерам строки и столбца (номера в исходной квадратной матрице; если соответствующий элемент лежит в одном из блоков, нужно выдать ссылку на ту переменную в составе двумерного динамического массива, в которой этот элемент хранится, иначе выдается ссылка на локальную статическую переменную, которая обнуляется при каждом вызове этой функции). Также написать функцию `destroy`, освобождающую память, выделенную функцией `create`.

Написать программу для вычисления экспоненты от комплексного аргумента, воспользовавшись тем, что $\exp(z) = \exp(x)(\cos(y) + i \sin(y))$.

Необходимо написать шаблон класса CalcElements в котором есть функция `add()` для выдачи суммы, `diff()` для выдачи разности, `div()` для выдачи частного и `prod()` для выдачи произведения элементов типа `int` или `float` . Также необходимо написать специализацию шаблона с функцией `concatenate()` для элементов типа `string` и `char`, чтобы объединить их в одну строку.

Написать класс `Vector` для реализации вектора. Координаты вектора должны являться закрытым полем данного класса. Выполнить перегрузку оператора `[]` — для индексации объектов. `()` — для присваивания координатам значений. Также реализовать перегрузку скалярного `*` и векторного `()` произведение векторов. Не забыть про метод для отображения координат вектора.