

Перебор, часть 2

Разбиение числа на слагаемые

Будем считать разбиения, отличающиеся только порядком слагаемых, одинаковыми.
Например, $1 + 7 + 9$ и $9 + 1 + 7$ одно и то же.

Пример:

Для числа 5 существует 7 разбиений:

$$5 = 1 + 1 + 1 + 1 + 1$$

$$5 = 1 + 1 + 1 + 2$$

$$5 = 1 + 1 + 3$$

$$5 = 1 + 2 + 2$$

$$5 = 1 + 4$$

$$5 = 5$$

Будем генерировать разбиения, в которых числа следуют в порядке не возрастания.

Разбиение числа на слагаемые

N – заданное число

sum – текущая сумма

rez – вектор, содержащий результат

$last$ – последняя цифра, добавленная в разбиение

partition (N , pos , sum , $last$)

ЕСЛИ $sum = N$

 Вывести первые pos значений вектора rez

ДЛЯ i от $last$ до ($N - sum$)

$rez[i] = i$

 partition(N , $pos + 1$, $sum + i$, i)

Разбиение числа на слагаемые

Задача

Разбиение числа на слагаемые. Выведите все разбиения числа n на натуральные слагаемые. Разбиения, отличающиеся только порядком слагаемых, считаются одинаковыми, поэтому выводите слагаемые в каждом разбиении в порядке не убывания.

Входные данные

Натуральное число n .

Выходные данные

Выведите разбиения числа n на слагаемые в лексикографическом порядке, каждое разбиение - в отдельной строке. Числа в каждом разбиении должны идти в порядке не убывания, разделяться знаками "+" (без пробелов) и в сумме давать n .

Пример входных данных

5

Пример выходных данных

1+1+1+1+1

1+1+1+2

1+1+3

1+2+2

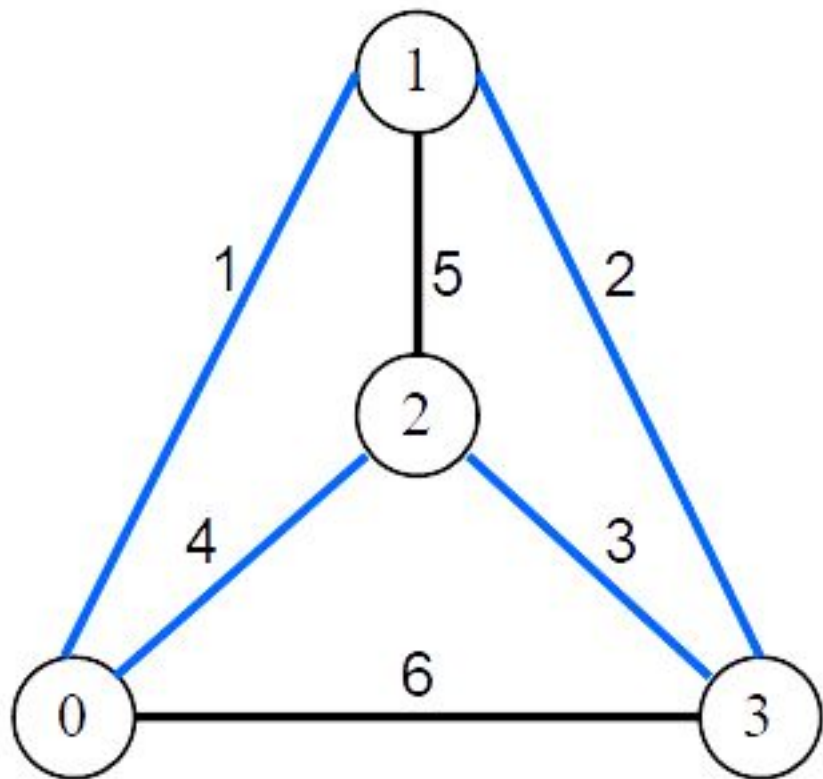
1+4

2+3

5

В качестве ответа на задание выберите разбиение на слагаемые числа $n = 7$ с номером 10 в

Задача коммивояжера



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

Задача коммивояжёра

N – количество городов

a – матрица путей

$city$ – вектор посещений (начальные значения нулевые)

way - текущий путь (содержит номера городов в порядке посещения)

len – длина текущего пути

cur – номер текущего города

$already$ – количество посещённых городов

$findWay(a, city, way, len, cur, already)$

ЕСЛИ $already = N - 1$

если существует путь в начальный город

вывести текущий путь и его длину ($len +$ длина последнего участка)

ДЛЯ i от 0 до N

ЕСЛИ $a[i][cur]$ не равно 0

если $city[i] = 0$

$city[i] = 1$

$way[already] = i$

$findWay(a, city, way, len + a[i][cur], i, already + 1)$

$city[i] = 0$

Задача коммивояжера

key = 0

min = 0

findWay(a, city, way, len, cur, already)

ЕСЛИ already = N - 1

ЕСЛИ существует путь в начальный город

ЕСЛИ key = 0

min = len + длина последнего участка

key = 1

ИНАЧЕ

ЕСЛИ (len + длина последнего участка) < min

min = len + длина последнего участка

ДЛЯ i от 0 до N

ЕСЛИ a[i][cur] не равно 0

ЕСЛИ city[i] = 0

city[i] = 1

way[already] = i

findWay(a, city, way, len + a[i][cur], i, already + 1)

city[i] = 0

Задача коммивояжера

Для самостоятельного решения

В первой строке задано натуральное число n - количество городов. Следующие n строк содержат длины дорог a_{ij} , по n чисел в каждой строке. Города пронумерованы числами от 0 до $n-1$. Гарантируется, что числа a_{ij} - натуральные, $a_{ij}=a_{ji}$ и $a_{ii}=0$ для i и j от 0 до $n-1$.

Выходные данные

В первой строке выведите одно целое число - минимальную длину пути коммивояжера. Во второй строке выведите последовательность из n чисел - сам путь. Путь должен содержать номера городов в порядке обхода и начинаться с номера 0.

Домашнее задание

Перебор правильных скобочных последовательностей с двумя типами скобок. Выведите все правильные скобочные последовательности с двумя типами скобок '(', ')', '[,]', содержащие $2n$ скобок, в лексикографическом порядке. В последовательности могут встречаться оба типа скобок или только один из них.

Входные данные

Натуральное число n .

Выходные данные

Выведите все правильные скобочные последовательности в лексикографическом порядке, каждую последовательность - в отдельной строке, без пробелов. Считайте, что '(' < ')' < '[' < ']'.
(Note: The original text contains a typo: '< ' < ']' which has been corrected to '[' < ']' in this translation.)

Тестовое задание

Найдите последовательность для $n = 7$ (состоящую из 14 скобок) с номером 8233 в лексикографическом порядке.

Домашнее задание

Легенда гласит, что Карл Фридрих Гаусс, учась в школе, смог быстро посчитать сумму целых чисел от 1 до 100, заметив, что $1 + 100 = 2 + 99 = \dots = 50 + 51$. Теперь решите задачу посложнее: можно ли перед каждым из чисел от 1 до N расставить знаки «+» или «-» так, чтобы сумма получившихся чисел была равна 0? Например, для $N = 3$ сумма $-1 -2 +3$ будет равна 0, а для $N = 2$ этого сделать нельзя.

Программа получает на вход целое неотрицательное число N , не превосходящее 105.

Программа должна вывести последовательность из N символов «+» или «-», соответствующих знакам, которые нужно расставить перед числами от 1 до N так, чтобы сумма получившихся чисел была равна 0. Если задача имеет несколько решений, нужно вывести один (любой) ответ. Если задача не имеет решения для данного N , нужно вывести одно слово «IMPOSSIBLE».

Примеры входных и выходных данных

Входные данные	Вывод программы	Примечание
3	---+	Вено также ++-
2	IMPOSSIBLE	