*Задачи динамического программирования*

1. *Проход по матрице штрафов*

Задана матрица натуральных чисел . За каждый проход через клетку матрицы с индексами  взимается штраф *.* Необходимо с минимальным суммарным штрафом пройти из первой клетки первой строки в клетку *(п,т).* Движение по матрице осуществляется следующим образом: из текущей клетки с координатами  можно перейти в любую из трех соседних клеток (если они существуют): . Найдите величину минимального суммарного штрафа.

1. *Проход по матрице платежей*

Задана матрица натуральных чисел . За каждый проход через клетку матрицы с индексами  выплачивается вознаграждение *.* Необходимо найти максимальную сумму, которую можно получить, если двигаться из клетки (1,1) в клетку *(п,т).* Движение по матрице осуществляется следующим образом: из текущей клетки с координатами  можно перейти в любую из двух соседних клеток (если они существуют): .

1. *Файловая система*

В файловой системе персонального компьютера файлы организованы в каталоги. В компьютере полное имя файла является строкой, состоящей из имен каталогов и имени файла, разделенных символом "\", причем "\" не может быть первым, последним символом, а также идти два раза подряд. Имя файла (каталога) может быть произвольной длины, но длина полного имени файла не может быть длиннее *п* символов. В качестве символов, допустимых к употреблению в именах файлов (каталогов), могут использоваться символы из алфавита, состоящего из  букв (символ "\" не входит в их число). Необходимо для данных   и   определить максимальное число файлов, которое можно записать на данный компьютер.

1. *Радиоактивные отходы*

При переработке радиоактивных материалов образуются отходы двух видов: опасные и неопасные. Для их хранения используются одинаковые контейнеры.

После помещения отходов в контейнеры, последние укладываются одной вертикальной стопкой. Стопка считается опасной, если в ней подряд идет не менее трех опасных контейнеров, в противном случае стопка считается безопасной. Для заданного количества контейнеров  определите число вариантов формирования безопасной стопки.

1. *Системы счисления*

Вычислите количество *п* -значных чисел в системе счисления с основанием *k*  таких, что их запись не содержит двух подряд идущих нулей.

1. *Локомотивы*

На железнодорожной колее размещены *п* локомотивов с номерам от 1 до *п, *. Они начинают двигаться в одну сторону, причем локомотив номер *i* движется со скоростью *i* м/с. Если локомотив, движущийся с большей скоростью, нагоняет более медленный локомотив, дальше они движутся один за другим со скоростью впереди идущего локомотива. Определите, сколько начальных расстановок дадут в результате ровно *k * групп движущихся локомотивов?

1. *Блок из единиц*

В таблице размера *п* на *т,* состоящей из 0 и 1, найдите квадратный блок максимального размера, состоящий только из 1.

1. *Обмен валют*

Матрица  заполнена неотрицательными вещественными числами. Число  определяет курс обмена валюты  на валюту . Так, например, если , то это значит, что за единицу валюты  дают 4 единицы валюты . Если , то полагаем, что курс обмена валюты  на валюту  прямо не установлен. Необходимо определить, можно ли, имея некоторую сумму денег в одной из валют, получить большую сумму денег в той же валюте, совершив несколько обменов.

1. *Лента*

Задана лента шириной в одну клетку и длиной в *п* клеток. На каждой клетке написано некоторое целое число. Играют два игрока, которые ходят поочередно. За один ход игрок отрезает от ленты одну из крайних клеток и забирает ее. Игра останавливается, когда лента заканчивается. Выигрыш игрока равен сумме чисел находящихся у него клеток ленты. Необходимо определить величину выигрыша, которую может гарантировать себе игрок, начинающий игру первым.

1. *Игра в числа*

Задается некоторое натуральное число *п <* 998. Двое играющих называют по очереди числа, меньшие 1000, по следующим правилам. Начиная с числа *п,* каждое новое число должно увеличивать одну из цифр предыдущего числа (возможно, незначащий нуль) на 1, 2 или 3. Проигравшим считается тот, кто называет число 999. Для заданного *п* необходимо определить, может ли выиграть игрок, делающий первый ход, при наилучших последующих ходах противника. В случае возможности выигрыша первым игроком, требуется напечатать все его возможные выигрышные первые ходы.

1. *Удали до палиндрома*

Вводится непустая строка, которая имеет длину не более 100 символов и состоит только из прописных латинских букв. Палиндромом называется такая строка символов, которая читается слева направо и справа налево одинаково. Необходимо удалить из строки минимальное количество символов так, чтобы получился палиндром. Этот палиндром нужно вывести на печать.

1. *Добавь до палиндрома*

Вводится непустая строка, которая имеет длину не более 100 символов и состоит только из прописных латинских букв. Палиндромом называется такая строка символов, которая читается слева направо и справа налево одинаково. Необходимо вставить в строку минимальное количество символов так, чтобы получился палиндром. Этот палиндром нужно вывести на печать.

1. *Последовательная упаковка дисков*

Имеется *п* дисков одинаковой толщины с радиусами . Эти диски упаковываются в коробку таким образом, что каждый из них стоит ребром на дне коробки, и все диски находятся в одной плоскости. Для последовательности дисков необходимо найти длину коробки, в которую все диски могут быть упакованы, при условии, что диски можно упаковывать только в порядке их следования.

1. *Произвольная упаковка дисков*

Имеется *п* дисков одинаковой толщины с радиусами . Эти диски упаковываются в коробку таким образом, что каждый из них стоит ребром на дне коробки, и все диски находятся в одной плоскости. Для последовательности дисков необходимо определить минимальную длину коробки, если порядок упаковки дисков в коробку может быть произвольным (указать при этом порядок, в котором упаковываются диски).

1. *Триангуляция многоугольника*

Выпуклый *п*-угольник () задается координатами своих вершин в порядке обхода по контуру. Необходимо разбить *п* -угольник на треугольники  диагоналями, не пересекающимися, кроме как в вершинах многоугольника, таким образом, чтобы сумма их длин была минимальной.

1. *Триангуляция многоугольника с минимаксной диагональю*

Выпуклый *п*-угольник () задается координатами своих вершин в порядке обхода по контуру. Необходимо разбить *п* -угольник на треугольники  диагоналями, не пересекающимися, кроме как в вершинах многоугольника, таким образом, чтобы максимальная из диагоналей имела наименьшую длину.

1. *Невозможная покупка*

Покупатель имеет *п* купюр достоинством  и продавец имеет *т* купюр достоинством *.* Необходимо найти максимальную стоимость товара, которую покупатель не может купить, потому что нет возможности точно рассчитаться за этот товар с продавцом, хотя денег на покупку у него достаточно.

1. *Точная сдача*

Покупатель имеет *п* купюр достоинством  и продавец имеет *т* купюр достоинством . Необходимо определить, может ли покупатель приобрести вещь стоимости  так, чтобы у продавца нашлась точная сдача (если она необходима).

1. *Железная дорога*

Железная дорога с односторонним движением имеет *п* станций. Известны цены билетов от *i* -ой станции до *j* -ой. Найдите минимальную стоимость проезда от начала до конца с учетом возможной экономии за счет пересадок.

1. *Возрастающая под последовательность без разрывов*

Необходимо из заданной числовой последовательности *а,* состоящей из *п* элементов, вычеркнуть минимальное количество элементов так, чтобы оставшиеся элементы образовали строго возрастающую под последовательность элементов. Построенный алгоритм должен иметь трудоемкость O(nlogn).

1. *Возрастающая подпоследовательность с разрывом*

Необходимо из заданной числовой последовательности *а,* состоящей из *п* элементов, вычеркнуть минимальное количество элементов так, чтобы в оставшейся подпоследовательности каждый последующий элемент был строго больше предыдущего, кроме не более одной пары соседних элементов (одного «разрыва»). Построенный алгоритм должен иметь трудоемкость O(nlogn).

1. *Выбор подпоследовательности*

Из элементов заданной последовательности *а* целых чисел длины *п* необходимо выбрать элементы таким образом, чтобы они образовали подпоследовательность наибольшей длины, в которой каждый последующий элемент делился бы нацело на предыдущий. Порядок следования элементов можно менять.

1. *Бинарная операция*

Задано конечное непустое множество с бинарной операцией □ (вообще говоря, некоммутативной и неассоциативной). Имеется *п* элементов  этого множества и еще один элемент *х*. Проверьте, можно ли так расставить скобки в выражении *а1 □ а2 □... □ ап,* чтобы в результате получился *х*. Построенный алгоритм должен иметь трудоемкость O(n3).

1. *Задача о рюкзаке*

Имеется *п* положительных чисел  - веса различных предметов, и число *т —* максимально допустимый вес рюкзака. Выясните, можно ли получить *т,* складывая некоторые из чисел . Построенный алгоритм должен иметь трудоемкость *О(пт).*

1. *Преобразование строк*

На вход подаются две символьные строки и *S1 и S2.* Необходимо преобразовать строку *S2* в строку *S2* с минимальным суммарным штрафом, который определяется следующим образом:

* штраф за удаление символа из строки *S2* равен *а* баллов;
* штраф за вставку символа в строку *S2 -b* баллов;
* штраф за выполнение замены символа в строке *S2* на любой другой символ - *с* баллов.