Дисциплина: «Теория алгоритмов»

**Лабораторная работа № 4**

**Динамическое программирование**

### Цель работы

Освоение методов динамического программирования для решения задач.

**Краткие теоретические сведения**

Динамическое программирование — это технология эффективной реализации рекурсивных алгоритмов посредством сохранения промежуточных результатов.

Секрет ее применения заключается в определении, выдает ли простой рекурсивный алгоритм одинаковые результаты для одинаковых подзадач.

Если выдает, то вместо повторений вычислений ответ каждой подзадачи можно сохранять в таблице для использования в дальнейшем, чем сэкономить время и ресурсы.

Основная идея состоит в том, чтобы:

- свести задачу для 𝑁 к задаче для чисел, меньших, чем 𝑁 (с помощью формулы)

- хранить все ответы в массиве

- заполнить начало массива вручную (для которых формула не работает)

- обойти массив и заполнить ответы по формуле

- вывести ответ откуда-то из этого массива

Чтобы решить задачу по динамике вы должны ответить на 5 вопросов:

Что лежит в массиве? (самый важный вопрос чаще всего)

Как инициализировать начало массива?

Как обходить массив? (чаще всего слева направо, но не всегда)

Какой формулой считать элементы массива?

Где в массиве лежит ответ?

Задачи динамического программирования делятся на два типа. *Подсчет количества вариантов решения*. В этом случае находят суммарное количество решений подзадач. *Поиск оптимального решения* (оптимизация целевой функции). В этом случае выбирают лучшее среди всех решений подзадач.

**Порядок выполнения лабораторной работы:**

1. Разработайте решение задачи в соответствии с вариантом с помощью методов динамического программирования
2. Протестируйте задачу на нескольких примерах .
3. Оформите отчет по лабораторной работе.
4. Защитите работу.

**Требования к оформлению лабораторной работы и отчета:**

1. Работа выполняется в форме консольного приложения
2. **Модуль программы** должен предваряться комментарием с текстом исходной задачи, имена переменных и компонентов должны быть понятны стороннему пользователю или сопровождаться комментарием, объясняющим их смысл. Имена проекта и модуля должны иметь осмысленные названия.
3. **Отчет по лабораторной работе** должен содержать: титульный лист, текст задания, описание принципа заложенного в решение задачи методом динамического программирования, листинг самой программы с комментариями.

Контрольные вопросы:

1) На какие типы делятся задачи, эффективно решаемые динамическим программированием?

2) Какой основной признак задачи говорит о том, что ее можно решить с помощью динамического программирования?

3) Какой из основных шагов решения задачи методом ДП говорит о том, что решение может быть применимо только к определенному типу задач?

**Задание**

|  |  |
| --- | --- |
| **Вариант 1** | |
| **Д**аны 3 целых натуральных числа, требуется подсчитать количество способов сформировать число N (вводится с клавиатуры), используя сумму данных трех чисел (допустимы повторы и использование не всех чисел). | |
| Пример:  числа: {1, 3, 5}  N=6 | 8 способов:  1+1+1+1+1+1  1+1+1+3  1+1+3+1  1+3+1+1  3+1+1+1  3+3  1+5  5+1 |

|  |  |
| --- | --- |
| **Вариант 2** | |
| Требуется подсчитать количество последовательностей длины N , состоящих из 0 и 1, в которых никакие две единицы не стоят рядом. | |
| Пример: N=4 | 8 вариантов  0000, 0001, 0010, 0100, 0101  1000, 1001, 1010 |

|  |  |
| --- | --- |
| **Вариант 3** | |
| Марио прыгает по трубам, расположенным на одной линии на равных расстояниях друг от друга. Трубы имеют порядковые номера от 1 до N . Вначале Марио стоит на трубе с номером 1. Он может прыгнуть вперед на расстояние от 1 до K труб, считая от текущей.  На каждой трубе Марио может получить или потерять несколько золотых монет (для каждой трубы это число известно). Определите, как нужно прыгать Марио, чтобы собрать наибольшее количество золотых монет.  Если правильных ответов несколько, выведите любой из них. Учитывайте, что Марио не может прыгать назад. | |
| Пример: N =5 K=3  0 2 -3 5 0 | Вывод:  7 — максимальное количество монет  3 — количество прыжков  1 2 4 5 — номера труб |

|  |  |
| --- | --- |
| **Вариант 4** | |
| Задано число N - длина последовательности (1 ≤ N ≤1000). Задана сама последовательность . Элементы последовательности - целые числа, не превосходящие 10000 по модулю.  Требуется вывести длину наибольшей строго возрастающей подпоследовательности и саму наибольшую возрастающую подпоследовательность. | |
| Пример: последовательность a[]={1 4 2 5 6 3 7} | Вывод:  5  1, 2, 5, 6, 7 |

|  |  |
| --- | --- |
| **Вариант 5** | |
| Дано N предметов массой m 1 , ..., m N и стоимостью c 1 , ..., c N соответственно.  Ими наполняют рюкзак, который выдерживает вес не более M. Какую наибольшую стоимость могут иметь предметы в рюкзаке? | |
| Пример:  Дано 4 предмета, имеющих параметры (масса, стоимость): {(2,7), (4,2), (1,5), (2,1)}.  Рюкзак выдерживает вес не более 6 кг. | Вывод:  5 — масса рюкзака  13 — максимальная стоимость |

|  |  |
| --- | --- |
| **Вариант 6** | |
| Дана строка, составленная из круглых, квадратных и фигурных скобок. Определите, какое наименьшее количество символов необходимо удалить из этой строки, чтобы оставшиеся символы образовывали правильную скобочную последовательность. Если возможных ответов несколько, выведите любой из них. | |
| Пример:  {([(]{)})] | Вывод:  [({})] |

|  |  |
| --- | --- |
| **Вариант 7** | |
| На День рождения Вам подарили сертификат на сумму N рублей в магазин спортивных товаров. У вас уже есть готовый список того, что вы хотите в нем купить, ранжированный по степени важности для Вас (макс.10). Найдите оптимальный список товаров, который возможно купить на сертификат и максимально быть довольным (список должен содержать не менее 10 позиций) при этом переплата должна отсутствовать. | |
| Пример:  сумма 2000  мяч: 500р. важность 7  кроссовки: 1500р. Важность 10  дождевик 300р важность 4  ... | Вывод:  итоговая сумма 2000  плащ+палатка  суммарная важность 17 |

|  |  |
| --- | --- |
| **Вариант 8** | |
| Вы приехали в Москву на 2 дня и хотите посетить максимальное количество достопримечательностей, у вас есть список из достопримечательностей, которые вы хотите посмотреть, оценка насколько сильно вы их хотите посмотреть (макс. 10) и примерное время, которое может быть затрачено на осмотр каждой достопримечательности.  Постройте оптимальный туристический маршрут на основе этого списка, чтобы ваша удовлетворенность была максимальной.  В списке должно быть не менее 10ти достопримечательностей. | |
| Пример:  Кремль — 0,5 дня; 10  Большой театр — 0,2 дня; 5  Третьяковская галерея — 1 день; 7  Зоопарк — 0,5 дня; 9  ... | Вывод:  Кремль  Третьяковская галерея  Зоопарк  Общая удовлетворенность 26 |

|  |  |
| --- | --- |
| **Вариант 9** | |
| Вы собираетесь в турпоход. Емкость вашего рюкзака 10кг. Вы составили список из предметов, которые можете взять в поход (не менее 10). У каждого предмета указаны вес и оценка важности (макс.10).  Вычислите как будет выглядеть оптимальный набор предметов для похода. | |
| Пример:  вода 3кг, 10;  книги 1 кг, 3;  еда 4кг, 9;  куртка 1кг, 6;  фотоаппарат 3кг, 7;  ... | Вывод:  куртка  книги  вода  еда  Общая важность 28 |

|  |  |
| --- | --- |
| **Вариант 10** | |
| На вершине лестницы, содержащей *N* ступенек, находится мячик, который начинает прыгать по ним вниз, к основанию. Мячик может прыгнуть на следующую ступеньку, на ступеньку через одну или через 2. (То есть, если мячик лежит на 8-ой ступеньке, то он может переместиться на 5-ую, 6-ую или 7-ую.) Определить число всевозможных "маршрутов" мячика с вершины на землю. | |
| Пример:  N=4 | Вывод:  7 |

|  |  |
| --- | --- |
| **Вариант 11** | |
| Вычислите n-й член последовательности, заданной формулами:  a2n= an+ an-1, a2n+1= an– an-1, a0= a1= 1. | |
| Пример:  N=4 | Вывод:  3 |

|  |  |
| --- | --- |
| **Вариант 12** | |
| В прямоугольной таблице NxM в начале игрок находится в левой верхней клетке. За один ход ему разрешается перемещаться в соседнюю клетку либо вправо, либо вниз (влево и вверх перемещаться запрещено). Посчитайте, сколько есть способов у игрока попасть в правую нижнюю клетку (N и M<10). | |
| Пример:  N=1 M=10 | Вывод:  1 |

|  |  |
| --- | --- |
| **Вариант 13** | |
| В каждой клетке прямоугольной таблицы 𝑁×𝑀записано некоторое число. Изначально игрок находится в левой верхней клетке. За один ход ему разрешается перемещаться в соседнюю клетку либо вправо, либо вниз (влево и вверх перемещаться запрещено). При проходе через клетку с игрока берут столько килограммов еды, какое число записано в этой клетке (еду берут также за первую и последнюю клетки его пути).  Требуется найти минимальный вес еды в килограммах, отдав которую игрок может попасть в правый нижний угол.  Вводятся два числа 𝑁 и 𝑀— размеры таблицы (1≤𝑁≤20,1≤𝑀≤20). Затем идет 𝑁 строк по 𝑀 чисел в каждой — размеры штрафов в килограммах за прохождение через соответствующие клетки (числа от 0 до 100). | |
| Пример:  5 5  1 1 1 1 1  3 100 100 100 100  1 1 1 1 1  2 2 2 2 1  1 1 1 1 1 | Вывод:  11 |

|  |  |
| --- | --- |
| **Вариант 14** | |
| На шахматной доске (8x8) стоит одна белая шашка. Сколькими способами она может пройти в дамки?  (Белая шашка ходит по диагонали. на одну клетку вверх-вправо или вверх-влево. Шашка проходит в дамки, если попадает на верхнюю горизонталь.)  Вводятся два числа от 1 до 8: номер номер столбца (считая слева) и строки (считая снизу), где изначально стоит шашка. | |
| Пример:  3 7 | Вывод:  2 |

|  |  |
| --- | --- |
| **Вариант 15** | |
| Дана прямоугольная доскаN×M (N строк и M столбцов). В левом верхнем углу находится шахматный конь, которого необходимо переместить в правый нижний угол доски. При этом конь может ходить ТОЛЬКО на две клетки вниз и на одну клетку вправо, либо на две клетки вправо и на одну клетку вниз (смотри рисунок).    Необходимо определить, сколько существует различных маршрутов, ведущих из левого верхнего в правый нижний угол. Вводятся два натуральных числаNиM(1≤ N, M ≤ 50). | |
| Пример:  4 4 | Вывод:  2 |

|  |  |
| --- | --- |
| **Вариант 16** | |
| Дана прямоугольная доска N × M (N строк и M столбцов). В левом верхнем углу находится шахматный конь, которого необходимо переместить в правый нижний угол доски. При этом конь может ходить только так, как показано на рисунке:    Необходимо определить, сколько существует различных маршрутов, ведущих из левого верхнего в правый нижний угол.  Вводятся два натуральных числа N и M (1 ≤ N, M ≤ 15). | |
| Пример:  7 15 | Вывод:  13309 |

|  |  |
| --- | --- |
| **Вариант 17** | |
| Узник пытается бежать из замка, который состоит из N×M квадратных комнат, расположенных в виде прямоугольника NxM. Между любыми двумя соседними комнатами есть дверь, однако некоторые комнаты закрыты и попасть в них нельзя. В начале узник находится в левой верхней комнате и для спасения ему надо попасть в противоположную правую нижнюю комнату. Времени у него немного, всего он может побывать не более, чем в N+M-1 комнате на своем пути, то есть перемещаться он должен только вправо или вниз. Определите количество маршрутов, которые ведут к выходу.  Входные данные  Первая строчка входных данных содержит натуральные числа N и M, не превосходящих 1000. Далее идет план замка в виде N строчек из M чисел в каждой. Одно число соответствует одной комнате: 1 означает, что в комнату можно попасть, 0 – что комната закрыта.  Выходные данные  Программа должна напечатать количество маршрутов, ведущих узника к выходу и проходящих через M+N-1 комнату, или слово Impossible, если таких маршрутов не существует. | |
| Пример:  3 5  1 1 1 1 1  1 0 1 0 1  1 1 1 1 1 | Вывод:  3 |

|  |  |
| --- | --- |
| **Вариант 18** | |
| В левом верхнем углу прямоугольной таблицы размером N×M находится черепашка. В каждой клетке таблицы записано некоторое число. Черепашка может перемещаться вправо или вниз, при этом маршрут черепашки заканчивается в правом нижнем углу таблицы.  Подсчитайте сумму чисел, записанных в клетках, через которую проползла черепашка (включая начальную и конечную клетку). Найдите наибольшее возможное значение этой суммы и маршрут, на котором достигается эта сумма.  Входные данные  В первой строке входных данных записаны два натуральных числа N и M, не превосходящие 100 — размеры таблицы. Далее идет N строк, каждая из которых содержит M чисел, разделенных пробелами — описание таблицы. Все числа в клетках таблицы целые и могут принимать значения от 0 до 100.  Выходные данные  Первая строка выходных данных содержит максимальную возможную сумму, вторая – маршрут, на котором достигается эта сумма. Маршрут выводится в виде последовательности, которая должна содержать N-1 букву D, означающую передвижение вниз и M-1 букву R, означающую передвижение направо. Если таких последовательностей несколько, необходимо вывести ровно одну (любую) из них. | |
| Пример:  5 5  9 9 9 9 9  3 0 0 0 0  9 9 9 9 9  6 6 6 6 8  9 9 9 9 9 | Вывод:  74  D D R R R R D D |

|  |  |
| --- | --- |
| **Вариант 18** | |
| Даны две последовательности, требуется найти длину их наибольшей общей подпоследовательности.  В первой строке входных данных содержится число N – длина первой последовательности (1 ≤ N ≤ 1000). Во второй строке заданы члены первой последовательности (через пробел) – целые числа, не превосходящие 10000 по модулю.  В третьей строке записано число M – длина второй последовательности (1 ≤ M ≤ 1000). В четвертой строке задаются члены второй последовательности (через пробел) – целые числа, не превосходящие 10000 по модулю.  Выходные данные  Требуется вывести одно число – длину наибольшей общей подпоследовательности двух данных последовательностей или 0, если такой подпоследовательности нет. | |
| Пример:  3  1 2 3  3  2 3 1 | Вывод:  2 |

|  |  |
| --- | --- |
| **Вариант 19** | |
| Дана текстовая строка. С ней можно выполнять следующие операции:  1. Заменить один символ строки на другой символ.  2. Удалить один произвольный символ.  3. Вставить произвольный символ в произвольное место строки.  Например, при помощи первой операции из строки "СОК" можно получить строку "СУК", при помощи второй операции - строку "ОК", при помощи третьей операции - строку "СТОК.  Минимальное количество таких операций, при помощи которых можно из одной строки получить другую, называется стоимостью редактирования или расстоянием Левенштейна.  Определите расстояние Левенштейна для двух данных строк.  Входные данные  Программа получает на вход две строки, длина каждой из которых не превосходит 1000 символов, строки состоят только из заглавных латинских букв.  Выходные данные  Требуется вывести одно число – расстояние Левенштейна для данных строк. | |
| Пример:  ABCDEFGH  ACDEXGIH | Вывод:  3 |

|  |  |
| --- | --- |
| **Вариант 20** | |
| Шаблоном называется строка, состоящая из английских букв (a, ..., z, A, ..., Z) и символов ? и \*. Каждый из символов ? разрешается заменить на одну произвольную букву, а каждый из символов \* – на произвольную (возможно пустую) последовательность букв. Про любую строку из букв, которую можно получить из шаблона такими заменами, будем говорить, что она удовлетворяет этому шаблону.  Имеются два шаблона. Требуется найти строку минимальной длины, которая удовлетворяет обоим шаблонам, либо выдать сообщение, что такой строки не существует.  Входные данные  Заданные шаблоны записаны в первых двух строках входных данных. Длина каждого шаблона не превосходит 80 символов.  Выходные данные  Выведите строку минимальной длины, удовлетворяющую обоим шаблонам, либо сообщение «No solution! | |
| Пример:  AB?  \*BC | Вывод:  ABC |

|  |  |
| --- | --- |
| **Вариант 21** | |
| Дана последовательность, требуется найти длину её наибольшей возрастающей подпоследовательности. Подпоследовательностью последовательности называется некоторый набор её элементов, не обязательно стоящих подряд.  Входные данные  В первой строке входных данных задано число N - длина последовательности (1 ≤ N ≤ 1000). Во второй строке задается сама последовательность (разделитель - пробел). Элементы последовательности - целые числа, не превосходящие 10000 по модулю.  Выходные данные  Требуется вывести длину наибольшей строго возрастающей подпоследовательности. | |
| Пример:  6  3 29 5 5 28 6 | Вывод:  3 |

|  |  |
| --- | --- |
| **Вариант 22** | |
| В некотором государстве в обращении находятся банкноты определенных номиналов. Национальный банк хочет, чтобы банкомат выдавал любую запрошенную сумму при помощи минимального числа банкнот, считая, что запас банкнот каждого номинала неограничен. Помогите Национальному банку решить эту задачу.  Входные данные  Первая строка входных данных содержит натуральное число N не превосходящее 100 — количество номиналов банкнот в обращении. Вторая строка входных данных содержит N различных натуральных чисел x1, x2, ..., xN, не превосходящих 106 — номиналы банкнот. Третья строчка содержит натуральное число S, не превосходящее 106 —сумму, которую необходимо выдать.  Выходные данные  Программа должна найти представление числа S виде суммы слагаемых из множества xi, содержащее минимальное число слагаемых и вывести это представление на экран (в виде последовательности чисел, разделенных пробелами). Если таких представлений существует несколько, то программа должна вывести любое (одно) из них. Если такое представление не существует, то программа должна вывести строку No solution. | |
| Пример:  5  1 3 7 12 32  40 | Вывод:  32 7 1 |

|  |  |
| --- | --- |
| **Вариант 23** | |
| Вы решили заказать пиццу с доставкой на дом. Известно, что для клиентов, сделавших заказ на сумму более 𝐶 рублей, доставка является бесплатной, при заказе на 𝐶 рублей и меньше доставка стоит B рублей.  Вы уже выбрали товар, стоимостью 𝐴 рублей. В наличии имеются еще 𝑁 товаров стоимостью 𝑑1, ..., 𝑑𝑁 рублей, каждый в единственном экземпляре. Их также можно включить в заказ.  Как потратить меньше всего денег и получить на дом уже выбранный товар в 𝐴 рублей?  Входные данные  Сначала вводятся числа 𝐴, 𝐶, 𝑁, а затем 𝑁 чисел 𝑑1, ..., 𝑑𝑁.  Все числа целые, 1 ≤ 𝐴 ≤ 1000, 1 ≤ 𝐶 ≤ 1000, 0 ≤ 𝑁 ≤ 1000, 1 ≤ 𝑑𝑖 ≤ 1 000 000.  Выходные данные  Выведите сначала суммарное количество денег, которое придется потратить. Если при этом вы планируете сделать дополнительный заказ c расчетом на бесплатную доставку, то далее выведите количество этих товаров и их номера в возрастающем порядке. | |
| Пример:  10 25  5  2 7 5 3 7 | Вывод:  26  3 1 2 5 |

|  |  |
| --- | --- |
| **Вариант 24** | |
| Палиндромом называется строка, которая одинаково читается как слева направо, так и справа налево. Подпалиндромом данной строки называется последовательность символов из данной строки (в том же порядке, но не обязательно идущих подряд), являющаяся палиндромом. Например, HELOLEH является подпалиндромом строки HTEOLFEOLEH. Напишите программу, находящую в данной строке подпалиндром максимальной длины.  на вход подается строка длиной не более 100 символов, состоящая из заглавных букв латинского алфавита.  Выведите длину максимального подпалиндрома. | |
| Пример:  THISISEASI | Вывод:  5 |

|  |  |
| --- | --- |
| **Вариант 25** | |
| Слово называется палиндромом, если его первая буква совпадает с последней, вторая – с предпоследней и т.д. Например: "abba", "madam", "x".  Для заданного числа K слово называется почти палиндромом, если в нем можно изменить не более K любых букв так, чтобы получился палиндром. Например, при K = 2 слова "reactor", "kolobok", "madam" являются почти палиндромами (подчеркнуты буквы, заменой которых можно получить палиндром).  Подсловом данного слова являются все слова, получающиеся путем вычеркивания из данного нескольких (возможно, одной или нуля) первых букв и нескольких последних. Например, подсловами слова "cat" являются слова "c", "a", "t", "ca", "at" и само слово "cat" (а "ct" подсловом слова "cat" не является).  Требуется для данного числа K определить, сколько подслов данного слова S являются почти палиндромами.  Входные данные  В первой строке вводятся два натуральных числа:N (1 ≤ N ≤ 5 000) – длина слова и K (0 ≤ K ≤ N).  Во второй строке содержится слово S, состоящее из N строчных латинских букв.  Выходные данные  Требуется вывести одно число – количество подслов слова S, являющихся почти палиндромами (для данного K). | |
| Пример:  5 1  abcde | Вывод:  12 |

|  |  |
| --- | --- |
| **Вариант 26** | |
| Шахматная ассоциация решила оснастить всех своих сотрудников такими телефонными номерами, которые бы набирались на кнопочном телефоне ходом коня. Например, ходом коня набирается телефон 340-49-27. При этом телефонный номер не может начинаться ни с цифры 0, ни с цифры 8.  Напишите программу, определяющую количество телефонных номеров длины N, набираемых ходом коня.  Входные данные  Вводится одно целое число N (1 ≤ N ≤ 20).  Выходные данные  Выведите  искомое количество телефонных номеров. | |
| Пример:  10 | Вывод:  11728 |