**Контрольная работа 2**

Решите одну или несколько задач с номерами N+12k, где N=№ вашего варианта, k=0,1,2,… . Чем больше решено задач, тем выше оценка на экзамене.

1. Написать алгоритм поиска *узла* с ключом, равным значению x: integer, в двоичном дереве поиска и *его* непосредственного предка. Известен корневой узел дерева.
2. Пусть известен *узел* в двоичном дереве поиска, имеющий двух непосредственных потомков и известен его непосредственный предок. Написать алгоритм удаления *этого узла.*
3. Найти в неупорядоченном двоичном дереве хотя бы один *элемент* с ключом, равным значению x: integer (если он существует), и *его* непосредственного предка. Известен корневой узел дерева.
4. Написать алгоритм построения простого однонаправленного циклического списка из n элементов. Число элементов n и значения ключей читаются из файла.
5. Простой однонаправленный список состоит из элементов с ключами типа integer. Преобразовать список так, чтобы элементы с максимальным и минимальным ключами стали соответственно первым и последним элементами списка. Известно начало списка.
6. Циклический однонаправленный список состоит из элементов с ключами типа integer. Все элементы списка упорядочены по ключам за исключением двух. Найти эти элементы, если известен некоторый текущий элемент списка.
7. Двоичное дерево поиска состоит из элементов с ключами типа integer, причем для каждого элемента (с ключом, равным t) все ключи в левом поддереве строго меньше t, а в правом поддереве все ключи больше либо равны t. *Ключ корневого узла стал равен произвольному значению x.* Написать алгоритм просеивания, преобразующий "испорченное" дерево к дереву поиска. Известен корневой узел дерева.
8. Дано двоичное дерево поиска. Написать алгоритм дополнения дерева буферным элементом. Известен корневой узел дерева.
9. Написать алгоритм удаления всех элементов с ключом, равным значению x: integer, в простом двунаправленном списке (отсортированном по возрастанию ключей). Известно начало списка.
10. Написать алгоритм вставки нового элемента с ключом, равным значению x: integer, в простой двунаправленный список (отсортированный по возрастанию ключей). Известно начало списка.
11. Написать алгоритм построения простого двунаправленного списка из n элементов. Число элементов n и значения ключей читаются из файла.
12. Написать процедуру инвертирования массива x из n элементов.
13. Написать процедуру циклического сдвига элементов массива x (из n элементов) на k позиций и подсчитать число обменов x(i) <-> x(j).
14. Написать процедуру бинарного поиска заданного целого значения в упорядоченном (по возрастанию или убыванию) массиве (x) целых чисел из n элементов. Подсчитать число сравнений.

**Динамическое программирование**

1. **Подматрица из единиц**

Вводится матрица *a(m,n)* из нулей и единиц. Найти в ней прямоугольную подматрицу из одних единиц максимального размера (т.е. с максимальным произведением высоты на длину).

1. **Проход по полю**

Задана матрица натуральных чисел *A(n,m).* За каждый проход через клетку *(i,j)* взимается штраф *A(i,j).* Необходимо минимизировать штраф и

а) Пройти из какой-либо клетки 1-й строки в n-ю строчку, при этом из текущей клетки можно перейти:

1) в любую из трёх соседних, стоящих в стpоке с номеpом на единицу большем;

2) в любую из восьми соседних клеток;

б) Реализовать пункт a) для перехода из клетки *(1,1)* в *(n,m).*

1. **Триангуляция**

Дан выпуклый n-угольник, n≥3, своим обходом по контуру. Разбить его на треугольники (n-3)-мя диагоналями, непересекающимися кроме как по концам, таким образом чтобы

а) сумма их длин была минимальной;

 б) максимальная из диагоналей имела наименьшую длину.

1. **Возрастающая подпоследовательность**

Из последовательности, состоящей из N чисел, вычеркнуть минимальное количество элементов так, чтобы оставшиеся образовали строго возрастающую последовательность.

1. Перемножение матриц

Пусть известно, что для перемножения матрицы размера n\*m на матрицу размера *mk* требуется *nmk* операций. Необходимо определить, какое минимальное число операций потребуется для перемножения n матриц *А1,...Аn*, заданных своими размерами *n(i)×m(i).* При этом можно перемножать любые две рядом стоящие матрицы, в результате чего получается матрица нужного размера.

*Замечание:*

*n(i)* – число строк в матрице *Ai,*

*m(i)* – число столбцов в матрице *Ai.*

1. Роман

Роман состоит из n глав, которые содержат *a1, a2, …, an* страниц. Требуется издать этот роман в *k* томах, чтобы самый толстый том содержал наименьшее количество страниц. Главы нельзя делить между томами.

**Структуры данных.**

1. ***Химия***

Задано N веществ и таблица их взаимодействий, то есть aij=0, если вещество с номером i не взаимодействует с веществом с номером j и aij=k (1 ≤ i,j,k ≤ N), если при их взаимодействии получается вещество с номером k. В пробирку одно за другим засыпаются вещества. Оказавшись рядом, они могут вступить в реакцию. Вновь образованное вещество, возможно, реагирует с нижележащим и так далее. Известно, что реакция взаимодействия происходит мгновенно и только между двумя соседними слоями. Описать алгоритм, определяющий по заданной последовательности, какие вещества останутся в пробирке.

1. ***Шахматы***

На шахматной доске отмечены две клетки. Используя очередь, определить, за какое наименьшее количество ходов можно добраться от первой отмеченной клетки до второй ходом шахматного коня.

1. ***Карточки***

Имеется n черных и белых карточек, сложенных в стопку. Карточки раскладываются на стол в одну линию следующим образом: первая кладется на стол, вторая под низ стопки, третья – на стол, четвертая – под низ стопки и так далее, пока все карточки не будут выложены на стол. Каким должно быть исходное расположение карточек в стопке, чтобы разложенные на столе карточки чередовались по цвету: белая, черная, белая, черная и так далее?

1. ***Мыши***

N серых и M белых мышей сидят по кругу. Кошка ходит по кругу по часовой стрелке и съедает каждую S -тую мышку. В первый раз счет начинается с серой мышки. Составить алгоритм, определяющий порядок, в котором сидели мышки, если через некоторое время осталось K серых и L белых мышей.

1. ***Куски бумаги***

Из листа клетчатой бумаги размером МN клеток удалили некоторые клетки. На какое количество кусков распадется оставшаяся часть листа?

Пример: если из шахматной доски удалить все клетки одного цвета, то оставшаяся часть распадется на 32 куска.

1. ***Цилиндр***

Лист клетчатой бумаги размером MN клеток склеили так, что образовался цилиндр высотой N клеток. Из полученного клетчатого цилиндра удалили несколько клеток. Определить, на какое количество кусков распадется оставшаяся часть цилиндра.

1. ***Простые множители***

Дано натуральное число K. Составить алгоритм нахождения первых K натуральных чисел (в порядке возрастания) таких, что в их разложение на простые множители входят только числа 2, 3 и 5.

**Алгоритмы на графах**

1. Задан набор неповторяющихся пар (Ai,Aj); Ai, Aj принадлежат множеству А={A1, A2, ..., An}. Необходимо составить цепочку максимальной длины по правилу (Ai,Aj)+(Aj,Ak)=(Ai,Aj,Ak). При образовании этой цепочки любая пара может быть использована не более одного раза.
2. Между N пунктами (N<=50) заданы дороги длиной A(i,j), где I,J-номера пунктов. Дороги проложены на разной высоте и пересекаются только в общих пунктах. В начальный момент времени из заданных пунктов начинают двигаться с постоянной скоростью M роботов (M=2 или 3), независимо меняя направление движения только в пунктах. Роботы управляются таким образом, чтобы минимизировать время до встречи всех роботов в одном месте. Скорость I-того робота может быть равна 1 или 2 . Остановка роботов запрещена.
3. Задание: написать программу, которая:

1) при заданных N,M и сети дорог единичной длины (все имеющиеся A(i,j)=1) определяет минимальное время, через которое может произойти встреча всех M роботов, при этом начальное положение роботов и скорость их движения известны.

2) Выполнить те же действия, что и в п. 1, но только для различных значений A(i,j).

*Примечание:* в случае невозможности встречи всех M роботов в одном месте ни в какой момент времени в результате выполнения программы должно быть сформировано соответствующее сообщение.

*Требования к вводу-выводу:*

1) Все входные данные - целые неотрицательные числа;

2) при задании сети дорог должно быть указано количество дорог – K и пункты их начала и конца в виде пар (i,j).

1. На плоскости расположено N точек. Имеется робот, который двигается следующим образом. Стартуя с некоторой начальной точки и имея некоторое начальное направление, робот движется до первой встреченной на его пути точки, изменяя в ней свое текущее направление на 90 градусов, т.е. поворачивая налево или направо. После этого он продолжает движение аналогично. Если робот достиг начальной точки, либо не может достичь новой точки (которую он еще не посещал), то он останавливается. Определить, может ли робот посетить все N точек, если:

1. Определены начальные точка и направление робота.

 2. Определена начальная точка, а направление робота можно выбирать.

3. Начальную точку и направление робота можно выбирать.

Координаты точек – целые числа, угол измеряется в радианах относительно оси ОХ.

1. **«ПУТЬ»**

Найти кратчайшее расстояние между двумя вершинами в графе. Найти все возможные пути между этими двумя вершинами в графе не пересекающиеся по

а) pебpам

б) вершинам.

1. Лабиринт задается матрицей смежности N\*N, где C(i,j)=1, если узел i связан узлом j посредством дороги. Часть узлов назначается входами, часть - выходами. Входы и выходы задаются последовательностями узлов X(1),..,X(p) и Y(1),..,Y(k) соответственно. Найти максимальное число людей, которых можно провести от входов до выходов таким образом, чтобы:

а) их пути не пересекались по дорогам, но могут пересекаться по узлам;

б) их пути не пересекались по узлам;

1. N шестеренок пронумерованы от 1 до N (N<= 10). Заданы M (0<=M<=45) соединений пар шестеренок в виде (i,j), 1<=i<j<=N (шестерня с номером i находится в зацеплении с шестерней j). Можно ли повернуть шестерню с номером 1? Если да, то найти количество шестерен, пришедших в движение. Если нет, то требуется убрать минимальное число шестерен так, чтобы в оставшейся системе при вращении шестерни 1 во вращение пришло бы максимальное число шестерен. Указать номера убранных шестерен (если такой набор не один, то любой из них) и количество шестерен, пришедших в движение.
2. Имеется N прямоугольных конвертов и N прямоугольных открыток различных размеров. Можно ли разложить все открытки по конвертам, чтобы в каждом конверте было по одной открытке. Замечание. Открытки нельзя складывать, сгибать и т.п., но можно помещать в конверт под углом. Например, открытка с размерами сторон 5:1 помещается в конверты с размерами 5:1, 6:3, 4.3:4.3, но не входит в конверты с размерами 4:1, 10:0.5, 4.2:4.2.
3. Составить программу для нахождения произвольного разбиения 20 студентов на 2 команды, численность которых отличается не более чем в 2 раза, если известно, что в любой команде должны быть студенты, обязательно знакомые друг с другом. Круг знакомств задается матрицей (20,20) с элементами

 A(ij)={1,если i студент знаком с j

 {0,иначе.

1. Имеется N человек и прямоугольная таблица А[1:N,1:N];элемент A[i,j] равен 1, если человек i знаком с человеком j, А[i,j] =А[j,i]. Можно ли разбить людей на 2 группы, чтобы в каждой группе были только незнакомые люди.
2. На олимпиаду прибыло N человек. Некоторые из них знакомы между собой. Можно ли опосредованно перезнакомить их всех между собой? (Незнакомые люди могут познакомиться только через общего знакомого).
3. Пусть группа состоит из N человек. В ней каждый имеет (N/2) друзей и не больше K врагов. У одного из них есть книга, которую все хотели бы прочитать и потом обсудить с некоторыми из остальных. Написать программу, которая:

1. Находит способ передачи книги таким образом, чтобы она побывала у каждого в точности один раз, переходя только от друга к другу и, наконец, возвратилась к своему владельцу.

2.Разбивает людей на S групп, где будет обсуждаться книга, таким образом, чтобы вместе с каждым человеком в ту же самую группу вошло не более P его врагов.

Примечание: предполагается, что S\*P>=K.

1. В заданном графе необходимо определить, существует ли замкнутая цепь, проходящая по каждому ребру графа ровно один раз. Постройте такую цепь.
2. N различных станков один за другим объединены в конвейер. Имеется N рабочих. Задана матрица C[N , N], где C[i,j] производительность i-ого рабочего на j-ом станке. Определить

а) на каком станке должен работать каждый из рабочих, чтобы производительность была максимальной;

б) то же, но станки расположены параллельно и выполняют однородные операции.

1. На плоскости задан граф с N вершинами. Количество ребер, соединенных с каждой вершиной, равно 3.

Пример:

 B┌────────────┐C

 │\ /│

 │ \ G F / │

 │ ┌──────┐ │

 │ │ │ │

 │ │ │ │

 │ └──────┘ │

 │ / H E\ │

 │ / \ │

 A└────────────┘D

Пусть вершины X,Y и Z являются соседями вершины Т. Будем считать, что Y левый, а Z –правый сосед вершины Т относительно вершины X, если ориентированный угол XTZ меньше ориентированного угла XTY (положительным будем считать направление против часовой стрелки). Например вершина Е является правым соседом вершины Н относительно А, а G - левым, поскольку ориентированный угол АНЕ меньше ориентированного угла AHG. (Ребра считаются отрезками). Составьте программу, которая:

1. Вводит координаты вершин графа и его ребра и рисует граф на экране компьютера, производя при этом подходящее масштабирование (ребра выводятся как отрезки).

2. Пусть заданы две начальные соседние вершины XO и X1 и последовательность вида LLRRL... Тогда программа находит путь на графе XOX1X2...Xn для вершин которого выполнено:

 -первые два являются заданными XO и X1

 -Xi+1 является левым или правым соседом Xi относительно Xi-1 в зависимости от заданной последовательности, при этом L означает левый, а R -правый.

 Пример: В заданном графе пусть даны начальные вершины А и H и последовательность LRRLLR. Тогда программа должна найти путь AHGFEDCB.

3. Рисует на экране путь,найденный в п.2.

4. Пусть даны начальная и конечная вершина. Программа должна найти путь,проходящий через минимальное число вершин, вывести его на экран и найти 2 первые вершины и управляющую последовательность для этого пути, как определено в п.2.

1. Имеется N городов. Для каждой пары городов (I,J) можно построить дорогу, соединяющую эти два города и не заходящие в другие города. Стоимость такой дороги A(I,J). Вне городов дороги не пересекаются. Написать алгоритм для нахождения самой дешевой системы дорог, позволяющей попасть из любого города в любой другой. Результаты задавать таблицей B[1:N,1:N], где B[I,J]=1 тогда и только тогда, когда дорогу, соединяющую города I и J, следует строить.
2. Вводится N – количество домов и К – количество дорог. Дома пронумерованы от 1 до N. Каждая дорога определяется тройкой чисел – двумя номерами домов – концов дороги и длиной дороги. В каждом доме живет по одному человеку. Найти точку – место встречи всех людей, от которой суммарное расстояние до всех домов будет минимальным. Если точка лежит на дороге, то указать номера домов – концов этой дороги и расстояние от первого из этих домов. Если точка совпадает с домом, то указать номер этого дома.

*Примечание*: длины дорог – положительные целые числа.

1. N колец сцеплены между собой (задана матрица A(n\*n), A(i,j)=1 в случае, если кольца i и j сцеплены друг с другом и A(i,j)=0 иначе). Удалить минимальное количество колец так, чтобы получилась цепочка.
2. Янка положил на стол N выпуклых K-гранников и N различных типов наклеек по K штук каждая. Ночью кто-то наклеил наклейки на грани, по одной на грань. Помогите Янке расставить многогранники так, чтобы наклейка каждого типа была видна ровно K-1 раз.
3. **«Задача Ларсона»**

Пусть G – конечный неориентированный связный граф. Предположим, что он представляет собой систему тоннелей, в которых может прятаться беглец. Группа из S полицейских, двигаясь по туннелям, стремится схватить этого беглеца, который может двигаться с любой скоростью, стремясь избежать поимки. Требуется определить минимальное количество полицейских S, гарантирующих поимку беглеца.

1. **Колье**

Ювелиру заказали изготовить колье, состоящее из зацепленных друг с другом золотых колец. Ювелир занумеровал кольца числами от 1 до N и стал изготавливать украшение, используя кольца по очереди – сначала первое, затем второе, и т.д. Каждый раз он делал в тетради пометку, к какому кольцу он зацеплял очередное кольцо. Таким образом, в тетради появилась N-1 запись: сначала там появилось число 1 (первое кольцо ни к чему не прицеплялось, а второе – к первому), затем одно из чисел 1 или 2, в зависимости от того, к какому кольцу прицеплялось второе кольцо, и т.д.

Когда колье было готово, заказчик передумал его покупать и потребовал, чтобы ювелир изготовил из него золотую цепь, отцепив некоторые кольца.

Напишите программу, которая по записям в тетради рассчитывает длину максимальной цепочки, которую можно получить из колье.

1. **Приватизация железных дорог**

В государстве из каждого города выходят ровно 3 железные дороги, ведущие в другие города. Дороги решили приватизировать две частные компании. По решению антимонопольного комитета запрещено все три дороги, выходящие из одного города, отдавать одной компании.

Напишите программу, которая по заданной схеме железных дорог рассчитывает одно из возможных распределений всех дорог между компаниями, или сообщает, что требование антимонопольного комитета невыполнимо.

1. **Плуг**

Преступники совершили ограбление банка и решили скрыться из города на тракторе, к которому был прицеплен плуг. Проезжая по дороге, преступники уничтожали (перепахивали) эту дорогу, чтобы полицейские, которые бросятся в погоню через некоторое время, не смогли идти по той же дороге.

Напишите программу, которая по заданной схеме дорог находит множество городов, в которых преступники смогут найти надежное убежище, т.е. таких, куда из начального города не останется ни одной неповрежденной дороги. Маршрут движения, естественно, преступники выбирают сами.

1. **Автобусные маршруты**

Пришедший к власти в городе Фишбурге мэр решил реформировать движение городских автобусов. Для этого он потребовал, чтобы ему предоставили информацию, по каким маршрутам ходят все автобусы. Информация о каждом маршруте состояла из перечня всех остановок вдоль следования данного автобуса по кругу.

Мэр приказал заменить все маршруты одним, но включающим в себя все переезды, какие и были раньше, причем, если между двумя остановками ходили K автобусов, то и новый маршрут должен был проходить между этими остановками K раз.

Напишите программу, которая по заданному множеству имеющихся в городе маршрутов находит один из возможных требуемых маршрутов.

1. **Второй по длине путь**

В заданном взвешенном графе найдите второй по длине путь между двумя заданными вершинами.

1. **Выборы в Анчурии**

Государство Анчурия разбито на квадратные избирательные округа, которые в совокупности образуют прямоугольник N×M. От каждого округа в парламент страны избирается один депутат. Если большинство избирателей округа – сторонники действующего президента, то в парламент избирается сторонник президента. Перед выборами был произведен социологический опрос, точно установивший количество избирателей, настроенных за и против президента. На основании этих данных президент хочет максимально увеличить количество своих сторонников в парламенте. Для этого он может укрупнить некоторые из избирательных округов, слив два соседних округа в один, от которого будут избираться два депутата в парламент.

Требуется написать программу, которая по известным данным социологического опроса предлагает вариант укрупнения округов для достижения вышеизложенных целей.

1. **Лягушка и комар**

Игра ведется на поле 8×8, называемом болотом, на котором некоторые клетки являются водной гладью, а некоторые – кочками. В начальный момент на одной из кочек сидит лягушка, а над одной клеткой летает комар. Лягушка и комар делают ходы по очереди (начинает лягушка). Ход лягушки заключается в перепрыгивании на любую кочку, находящуюся в той же горизонтали или вертикали, где в данный момент находится лягушка. Лягушка, в частности, может остаться на месте. Ход комара заключается в обязательном перелете в любую соседнюю клетку. Если во время хода лягушка перелетает над клеткой, где в данный момент находится комар, она его съедает, и игра заканчивается ее победой. После победы лягушке разрешается приземлиться в любую клетку болота (не обязательно на кочку).

Требуется написать программу, которая по заданному расположению кочек на болоте и начальным положениям лягушки и комара, рассчитывает ход лягушки, ведущей ее к победе за наименьшее количество ходов, независимо от игры комара. Если у лягушки нет выигрышной стратегии, то программа должна это сообщить.