**Задания**

Сформулируйте заданную проблему как проблему удовлетворения ограничений. Определите переменные, домены и ограничения. Создайте модель. Решите задачу в MiniZinc.

**Задание 1.**

Есть 12 задач (с номерами 1..12) и 6 рабочих (с номерами 1..6). Следующий массив переменных 0..1

array [1..6,1..12] of var 0..1:Assign ;

представляет решения о назначении работников задачам. Assign[w, t] = 1, если рабочий w назначен на задачу t, иначе Assign[w, t] = 0.

**Первое подзадание**

Напишите модель, которая гарантирует, что на каждую задачу назначается как минимум 2 работника.

**Второе подзадание**

Расширьте свою модель, чтобы убедиться, что ни одному работнику не назначено более 5 задач.

**Третье плдзадание**

Определенные пары рабочих не могут быть назначены на одну и ту же задачу. В следующей матрице

array [1..6,1..6] of 0..1: bad\_pair =

[|0,1,0,1,0,0

|1,0,0,0,1,0

|1,0,0,1,0,0

|0,0,1,0,1,0

|0,0,1,0,0,0

|0,0,0,1,0,0 |] ;

*bad*\_*pair*[*w*1*,w*2] = 1 означает, что рабочие w1 и w2 не могут быть назначены на одну и ту же задачу.

Расширьте свою модель второй задачи, чтобы удовлетворить ограничение плохой пары.

**Задание 2.**

У вас есть сумка, которая может нести 20 кг. У вас есть набор вещей, которые вы хотите взять с собой, и их вес:

enum items = {book, jacket, washbag, computer, boots,

alarmclock, anorak, food} ;

array [items] of int: weight = [2,4,3,8,7,1,2,6] ;

Первое подзадание

Эти предметы имеют для вас определенную ценность:

array [items] of int: value = [6,10,8,25,22,4,5,20] ;

Упакуйте предметы, которые вы можете носить в своей сумке, которые принесут вам максимально возможную общую ценность.

Второе подзадание

Рюкзак также имеет ограниченную вместимость, а общий объем предметов, которые он может поместить внутрь, составляет 2000 см2. Каждый предмет имеет не только вес, но и объем:

array [items] of int: volume =

[250, 500, 300, 250, 650, 130, 150, 600] ;

Найдите лучшее решение, как и в первой подзадаче, но общий объем предметов в рюкзаке не может быть больше вместимости рюкзака.

Третье подзадание

Некоторые вещи в сочетании стоят больше, а некоторые меньше. Вот дополнительный (или уменьшенный) балл, который вы получаете за каждую пару:

array [items,items] of int: extra\_value =

[| 0, 0, 0,-5, 0, 0, 0, 0

| 0, 0, 0, 0, 3, 0,-2, 0

| 0, 0, 0, 0, 0, 0, 0, 0

|-5, 0, 0, 0, 0,-2, 0, 0

| 0, 3, 0, 0, 0, 0, 0, 0

| 0, 0, 0,-2, 0, 0, 0, 0

| 0,-2, 0, 0, 0, 0, 0, 0

| 0, 0, 0, 0, 0, 0, 0, 0

|] ;

Если extra\_value[i1, i2] = 3 и предметы i1 и i2 находятся в вашей сумке, то общая стоимость вашей сумки увеличивается на 3. Естественно, если extra\_value[i1, i2] = −2, то она уменьшается на 2.

Расширьте свою модель для первой подзадачи, чтобы максимизировать общую сумму с измененными значениями.

**Задание 3.**

Есть два склада, A и B, каждый со своим грузовиком (truckA и truckB).

Каждый склад имеет четырех клиентов: A1,A2,A3,A4 и B1,B2,B3,B4.

Имеется таблица расстояний между каждой парой покупателей и между складами и каждым покупателем.

Dist =

% A A1 A2 A3 A4 B B1 B2 B3 B4

[| 0, 160, 150, 590, 340, 650, 725, 560, 350, 200 % A

|160, 0, 260, 680, 280, 650, 820, 715, 500, 150 % A1

|150, 260, 0, 440, 260, 520, 620, 490, 280, 150 % A2

|590, 680, 440, 0, 490, 240, 390, 435, 400, 550 % A3

|340, 280, 260, 490, 0, 660, 800, 700, 510, 140 % A4

|650, 650, 520, 240, 660, 0, 160, 250, 340, 670 % B

|725, 820, 620, 390, 800, 160, 0, 210, 380, 780 % B1

|560, 715, 490, 435, 700, 250, 210, 0, 215, 650 % B2

|350, 500, 280, 400, 510, 340, 380, 215, 0, 450 % B3

|200, 150, 150, 550, 140, 670, 780, 650, 450, 0 |] ; % B4

Предполагая, что грузовики имеют постоянную скорость, то элементы таблицы также можно понимать как время, необходимое для проезда между пунктами.

Первое подзадание

Грузовик А стартует со склада А; обслуживает клиентов склада А; и он возвращается на склад A. Аналогично для грузовика B.

Задача состоит в том, чтобы свести к минимуму время, когда последний грузовик возвращается на склад.

Второе подзадание

У каждого клиента есть временное окно, в течение которого его необходимо посетить, указанное здесь:

TimeWindow =

[|0, 2000 %A

|800,1800 %A1

|100, 600 %A2

|200,1500 %A3

|0, 500 %A4

|0, 2000 %B

|200,1700 %B1

|100,1200 %B2

|600,2500 %B3

|300,1700 %B4

|] ;

Например, клиента A1 нужно посетить в момент T, где 800 ≤ T ≤ 1500. Требование такое же, как и в первой подзадаче, но дополнительно удовлетворяющее ограничениям временного окна.

Третье подзадание

Предположим, что грузовик А может обслужить любого клиента, и грузовик Б тоже может это сделать. Клиенты должны быть посещены в пределах их временных окон. Грузовик А по-прежнему должен начинать и заканчивать на складе А, и аналогичным образом грузовик Б должен начинать и заканчивать на складе Б.

Задача состоит в том, чтобы свести к минимуму время, когда последний грузовик возвращается на склад.

**Задание 4.**

Туристическая компания проводит несколько туров в течение сезона, и для каждого тура нужен гид. Каждый тур имеет день начала и место, продолжительность (дни) и место окончания.

Любой экскурсовод может вести любую экскурсию, но только по одну! После того, как гид возглавит какую-либо экскурсию, он должен быть нанят на весь сезон (при стандартной стоимости 10000). Когда гид заканчивает одну экскурсию и начинает другую, необходимо оплатить проезд от конечной точки первой экскурсии до начальной точки следующей. На каждую экскурсию должен быть назначен ровно один гид.

Входными данными является набор туров, требующих гида, и список транспортных расходов по местоположению. Результатом является набор маршрутов — по одному для каждого активного гида — и общая стоимость. Параметры указаны следующим образом:

% Cost for each active tour guide

int: tour\_guide\_cost = 10000 ;

enum locations ;

% Each pair of locations has a travel cost recorded as an

integer

array [locations, locations] of int: travel\_cost ;

% The total number of planned tours

int: tour\_ct ;

set of int: all\_tours = 1..tour\_ct ;

% Each tour has a start day, duration, start location and end

location

array [all\_tours] of int: tour\_start;

array [all\_tours] of int: tour\_dur;

array [all\_tours] of locations: tour\_start\_loc;

array [all\_tours] of locations: tour\_end\_loc;

Возможный файл данных, содержащий значения параметров:

% Example data for a toy problem instance

locations = {rome, paris, prague, munich, vienna, end} ;

travel\_cost =

% Rome, Paris, Prague, Munich, Vienna, End

[| 0, 1106, 923, 699, 764, 0 % Rome

| 1106, 0, 886, 685, 1034, 0 % Paris

| 923, 866, 0, 300, 251, 0 % Prague

| 699, 685, 300, 0, 355, 0 % Munich

| 764, 1034, 251, 355, 0, 0 % Vienna

| 0, 0, 0, 0, 0, 0 % End

|] ;

tour\_ct = 7 ;

tour\_start = [20,25,30,40,43,50,100] ;

tour\_dur = [15, 15, 15, 12, 10, 8, 0] ;

tour\_start\_loc = [paris, paris, paris, paris, munich, munich,

end] ;

tour\_end\_loc = [rome, rome, prague, munich, vienna, munich,

end] ;

Стоимость услуг гида взимается только с гидов, которые проводят хотя бы одну экскурсию, и каждый такой гид ведет серию экскурсий. Гид должен ехать от конечной точки текущего тура до места начала следующего в его последовательности туров, стоимость переезда, указана в Матрице travel\_cost. Естественно, каждый тур в последовательности должен заканчиваться до следующего.

В каждом туре должен быть гид.

Задача состоит в том, чтобы назначить гида для каждого тура, минимизировав расходы на гидов задание плюс их транспортные расходы.

**Задание 5.**

Вы должны расставить числа 1..n по порядку, чтобы сумма разностей соседних чисел была максимальной.

Например, если n равно 4, то сумма разностей

[1,2,3,4]

равно 3. Однако максимальная сумма разностей достигается на порядке

[2,4,1,3]

В этом случае сумма разностей равна 7. Обратите внимание, что существует еще одна последовательность этих чисел с такой же суммой разностей.)

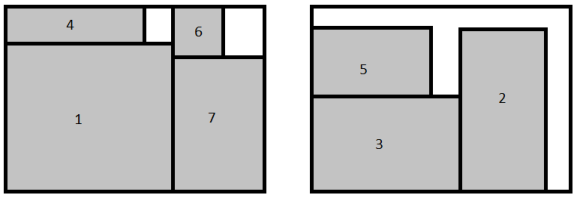
**Задание 6.**

Двумерная бин-упаковка.

Вам дан набор прямоугольных элементов, где элемент i имеет ширину wi и высоту hi. Задача состоит в том, чтобы упаковать все предметы в одинаковые ящики шириной W и высотой H, используя как можно меньше ящиков.

Перекрытие предметов не допускается, а также разрезание или вращение предметов.

Рисунок ниже иллюстрирует идею:



Решите двумерную задачу упаковки контейнеров, используя MiniZinc.

**Задание 7. (018)**

Вам дается 8-литровое ведро воды и два пустых ведра, которые могут содержать 5 и 3 литров соответственно. Вы должны разделить воду на две части, разлив воду между ведрами (то есть, чтобы в итоге получилось 4 литра в 8-литровом ведре и 4 литра в 5-литровом ведре).

Какое минимальное количество переливаний воды между ведрами?

**Задание 8.**

В задаче *о расположении склада* компания рассматривает возможность открытия складов в некоторых местах-кандидатах, чтобы снабжать свои существующие магазины. Каждый возможный склад имеет одинаковую стоимость обслуживания и вместимость, определяющую максимальное количество магазинов, которые он может обслуживать. Каждый магазин должен снабжаться ровно одним открытым складом.

Стоимость поставки в магазин зависит от склада. Цель состоит в том, чтобы определить, какие склады открыть и какие из этих складов должны снабжать различные магазины, чтобы сумма затрат на обслуживание и снабжение была минимизирована.

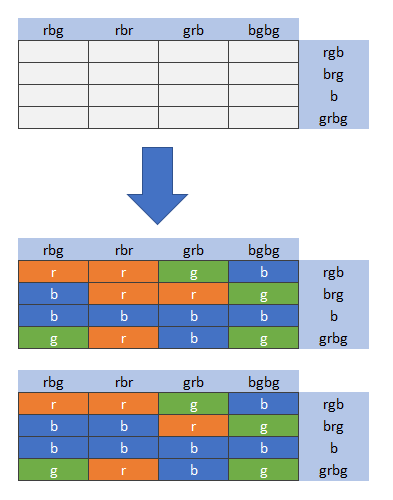
В качестве примера рассмотрим следующие данные:

**Задание 9.**

В задаче совмещения фотографий группа людей должна выстроиться в очередь для фотографирования. Каждый человек может дать предпочтения, с кем стоять рядом на фотографии. Цель состоит в том, чтобы разместить людей так, чтобы было удовлетворено как можно больше предпочтений.

**Задание 10.**

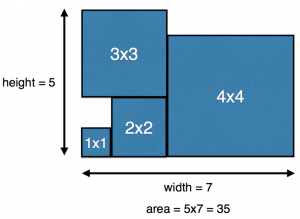
**Описание головоломки** : игроку дается пустое игровое поле (разного размера), которое он должен заполнить *n* -цветами, используя шаблоны подсказок для строк и столбцов. Каждый образец подсказки представляет собой последовательность цветов в этой строке / столбце, но с удаленными последовательными дубликатами.

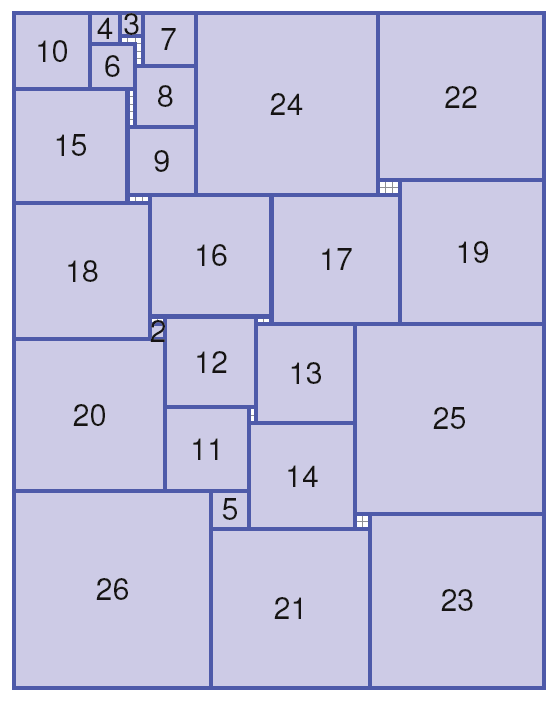


Выдать все правильные варианты заполнения игрового поля.

**Задание 11.**

Задача состоит в том, чтобы упаковать все квадраты размеров от 1×1 до NxN в наименьший возможный прямоугольник, т.е. в прямоугольник с минимальной площадью.



[](https://hsimonis.files.wordpress.com/2010/07/squares26.png)

Оптимальное решение для N=26

**Задание 12.**

Мы собираемся на ужин. На ужине может присутствовать от 1 до 6 бабушек и дедушек, от 1 до 10 родителей и от 1 до 40 детей. За дедушку (бабушку) нужно платить 3 евро за ужин, за родителей 2 евро и за детей 0,50 евро. Всего за ужином должно быть 20 человек, и он должен стоить ровно 20 евро. Задача, которую необходимо решить, состоит в том, чтобы узнать, сколько бабушек и дедушек, родителей и детей собирается обедать.

**Задание 13.**

Сможете ли вы найти возраст по следующему диалогу?

Алекс: Сколько тебе лет, мама?

Мама: Наши три возраста в сумме составляют ровно семьдесят лет.

Алекс: А сколько тебе лет, папа?

Папа: Всего в шесть раз старше тебя, сын мой.

Алекс: Буду ли я когда-нибудь вдвое моложе тебя, папа?

Папа: Да Алекс; и когда это произойдет, сумма наших трех возрастов будет ровно в два раза больше, чем сегодня.

Напишите программу, чтобы решить эту головоломку и вернуть возраст Алекса, Мамы и Папы на сегодня.

Подсказка: моделируйте возраст в месяцах.

**Задание 14.**

«Магический квадрат» — это прямоугольный массив различных чисел, обычно от 1 до n2, такой, что каждый столбец, строка и обе диагонали имеют одинаковую сумму. Постоянная сумма в каждой строке, столбце и диагонали называется магической суммой, M. Решите головоломку для магического квадрата с n = 4 и M = 34.

**Задание 15.**

Я знаю трех человек: Пэт, Куинси и Рэй. Мне нравится по крайней мере один из них. Если мне нравится Пэт, но не Рэй, тогда мне нравится и Куинси. Мне нравятся и Рэй, и Куинси, или ни один из них. Опишите факты как логические ограничения и определите, кто мне точно нравится, а кто может не нравиться.

**Задание 16.**

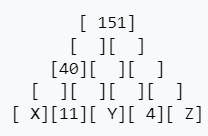
Решите следующую криптоарифметическую задачу. Замените разные буквы разными числами так, чтобы каждое число было абсолютной разностью двух чисел ниже (например, A = |B - C|), а числа были целыми положительными числами от 1 до 10.



**Задание 17.**

Треугольник Паскаля/Головоломка

Эта головоломка включает в себя треугольник Паскаля, также известный как пирамида чисел.



Каждый кирпич пирамиды представляет собой сумму двух кирпичей, расположенных под ним.

Из трех недостающих чисел в основании пирамиды среднее является суммой двух других (то есть Y = X + Z).

Напишите программу для решения этой головоломки.

**Задание 18.**

Проблема в том, чтобы запланировать турнир для n команд на n - 1 неделю, причем каждая неделя делится на n/2 периодов, и каждый период делится на два слота. Первая команда в каждом слоте играет дома, а вторая играет с первой командой на выезде. Турнир должен удовлетворять следующим трем ограничениям: каждая команда играет один раз в неделю; каждая команда играет не более двух раз за один и тот же период турнира; каждая команда играет с каждой другой командой.

Пример расписания для 8 команд:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 неделя | Неделя 2 | Неделя 3 | Неделя 4 | Неделя 5 | 6 неделя | Неделя 7 |
| Период 1 | 0 v 1 | 0 v 2 | 4 v 7 | 3 v 6 | 3 v 7 | 1 v 5 | 2 v 4 |
| Период 2 | 2 v 3 | 1 v 7 | 0 v 3 | 5 v 7 | 1 v 4 | 0 v 6 | 5 v 6 |
| Период 3 | 4 v 5 | 3 v 5 | 1 v 6 | 0 v 4 | 2 v 6 | 2 v 7 | 0 v 7 |
| Период 4 | 6 v 7 | 4 v 6 | 2 v 5 | 1 v 2 | 0 v 5 | 3 v 4 | 1 v 3 |

**Задание 19.**

**Проблема планирования репетиций**

Концерт должен состоять из девяти музыкальных произведений разной продолжительности, каждое из которых включает разное сочетание пяти участников оркестра.

Игроки могут приходить на репетиции непосредственно перед первым произведением, в котором они участвуют, и уходить сразу после последнего произведения, в котором они участвуют. Проблема состоит в том, чтобы разработать порядок, в котором можно репетировать пьесы, чтобы свести к минимуму общее время, в течение которого игроки ждут начала исполнения, т. е. общее время, когда игроки присутствуют, но не играют в данный момент.

В таблице ниже 1 указывает, что игрок требуется для соответствующей репетиции, в противном случае 0. Продолжительность (т.е. время, необходимое для репетиции каждой пьесы) выражается в некоторых неуказанных единицах времени.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Кусок | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| Игрок 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 |
| Игрок 2 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 |
| Игрок 3 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| Игрок 4 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| Игрок 5 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 |
| Продолжительность | 2 | 4 | 1 | 3 | 3 | 2 | 5 | 7 | 6 |

Например, если бы девять пьес репетировались в порядке нумерации, как указано выше, то общее время ожидания составило бы:

Игрок 1: 1+3+7=11

Игрок 2: 1+5=6

Игрок 3: 1+3+3+2=9

Игрок 4: 4+1+3+5+7=20

Игрок 5: 3

всего 49 единиц. Оптимальная последовательность дает 17 единиц времени ожидания.

**Задание 20.**

### **Проблема распределения талантов**

Проблема возникает при составлении графика съемок фильма. В разные дни съемок требуются разные группы актеров, и актерам платят за дни, которые они проводят в ожидании на съемочной площадке. Единственная разница между проблемой планирования талантов и проблемой репетиций заключается в том, что разным актерам платят по разным ставкам, так что стоимость времени ожидания зависит от того, кто ждет. Цель состоит в том, чтобы свести к минимуму общие затраты на оплату ожидания актеров.

*Film1*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| День | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | Стоимость/100 |
| Актер 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 10 |
| Актер 2 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 4 |
| Актер 3 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 5 |
| Актер 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5 |
| Актер 5 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 5 |
| Актер 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 40 |
| Актер 7 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 4 |
| Актер 8 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 20 |
| Продолжительность | 2 | 1 | 1 | 1 | 1 | 3 | 1 | 1 | 1 | 2 | 1 | 1 | 2 | 1 | 2 | 1 | 1 | 2 | 1 | 1 |  |

*Film2*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| День | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | Стоимость/100 |
| Актер 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 40 |
| Актер 2 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 20 |
| Актер 3 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 20 |
| Актер 4 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 10 |
| Актер 5 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 5 |
| Актер 6 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 10 |
| Актер 7 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 5 |
| Актер 8 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 4 |
| Актер 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 5 |
| Актер 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 4 |
| Продолжительность | 1 | 1 | 1 | 1 | 3 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  |

 Минимальная стоимость времени ожидания для задачи планирования талантов *в фильме «Фильм 1»* составляет 14 600, а в случае *«Фильма 2»* — 8 700 (в единицах, используемых в таблице — обратите внимание, что стоимость в таблице равна /100).

**Задание 21.**

Задача распределения танков заключается в назначении различных грузов (объемов химических продуктов, подлежащих отгрузке судном) в имеющиеся танки судна. Планы погрузки наливных судов, как правило, составляются планировщиками судов вручную, хотя создать высококачественные решения сложно. Ограничения, которые необходимо удовлетворить, в основном являются ограничениями сегрегации:

1. Предотвратите загрузку химикатов в определенные типы резервуаров, потому что:
   * Может потребоваться регулирование температуры химиката, а резервуар должен быть оборудован системой обогрева.
   * Резервуар должен быть устойчив к химикатам.
   * Танк все еще может быть загрязнен предыдущими грузами, несовместимыми с химическим веществом.
2. Предотвратить размещение нескольких пар грузов рядом друг с другом: необходимо учитывать не только химическое взаимодействие между различными грузами, но и температуру, при которой их необходимо транспортировать. Слишком разные температурные требования для соседних танков приводят к затвердеванию второго из-за охлаждения первым грузом или первый может стать химически неустойчивым из-за нагрева второго груза.

Чтобы свести к минимуму затраты и неудобства, связанные с очисткой резервуаров, идеальный план загрузки должен максимально увеличивать общий объем неиспользуемых резервуаров (т. е. свободное пространство).

В следующей модели вводится два семейства переменных:

1. Одна переменная X[t] для каждого резервуара t, представляющая тип продукта, который будет помещен в этот резервуар. Начальный домен — 0…P (некоторые резервуары могут быть недоступны для некоторых продуктов), 0 представляет собой фиктивный продукт, означающий, что продукт не назначен резервуару t (в этом случае X[t]=0).

2. Одна переменная загрузки L[p] для каждого продукта 0..P, загружаемого на судно. Эта переменная показывает, какой объем резервуара доступен/назначен для каждого продукта. Начальная нижняя граница этой переменной L[p] — это количество загружаемого продукта p. Это гарантирует наличие достаточного места для размещения этого продукта p (L[0]>=0).

Эти два набора переменных связаны с ограничением упаковки в бункер (X,caps,L), где c — вектор вместимости каждого бака. Это ограничение обеспечивает L[p] = sumt in tanks (X[t]=p)∗capa[t]∀p

Некоторые табличные ограничения могут быть добавлены к некоторым парам переменных соседних резервуаров x[i],x[j], чтобы избежать расположения несовместимых продуктов рядом друг с другом.

Цель состоит в том, чтобы максимизировать пустой объем, т.е. переменную L[0].

**Задание 22.**

В задаче SONET нам дан набор узлов, и для каждой пары узлов задан спрос ( количество каналов, необходимых для передачи сетевого трафика между двумя узлами). Спрос может быть нулевым, и в этом случае два узла не нужно соединять.

Кольцо SONET соединяет набор узлов. Узел устанавливается на кольце с помощью оборудования, называемого мультиплексором ввода-вывода (ADM). Каждый узел может быть установлен более чем в одном кольце. Сетевой трафик может передаваться от одного узла к другому только в том случае, если они оба установлены на одном кольце. Каждое кольцо имеет верхний предел количества узлов и предел количества каналов. Спрос пары узлов может быть разделен между несколькими кольцами.

Цель состоит в том, чтобы свести к минимуму общее количество используемых ADM при удовлетворении всех требований.

О модели

Модель проблемы имеет количество колец («r»), количество узлов («n»), «спрос» (какие узлы требуют связи) и пропускная способность узла каждого кольца («capacity\_nodes») в качестве параметров.

Назначение узлов кольцам моделируется двумерной матрицей « Rings», индексируемой количеством колец и узлов. Матрица-домен является булевой:

Если узел в столбце j назначен кольцу в строке i, то Rings[i,j] = 1

и 0 в противном случае. Таким образом, все «1» в «Rings» матрицы обозначают ADM.

Следовательно, цель состоит в том, чтобы минимизировать сумму по всем столбцам и строкам матричных «Rings».

**Задание 23.**

Дискретная задача определения размера партии и планирования (DLSP) — это задача производственного планирования, состоящая из определения графика производства с минимальными затратами (производственные затраты, затраты на настройку, затраты на переналадку, затраты на хранение и т. д.), чтобы не нарушались ограничения производительности оборудования и спрос на все товары удовлетворен. Горизонт планирования дискретен и конечен.

Описанный здесь вариант используется для экспериментов в [The StockingCost Constraint](https://www.csplib.org/Problems/prob058/data/stockingCostConstraint.pdf) .

Это многокомпонентная задача с одной машиной, производительность которой ограничена одной машиной за период. Существуют затраты на хранение и затраты на перенастройку, зависящие от последовательности, с соблюдением неравенства треугольника. Каждый заказ, состоящий из одной единицы определенного товара, имеет срок выполнения и должен быть изготовлен не позднее установленного срока. Стоимость хранения (запасов) заказа пропорциональна количеству периодов между датой выполнения и периодом производства. Стоимость перехода gi,j индуцируется при переходе от производства изделия *i* к другому j с gi,i =0,∀i. Здесь отставание не допускается. Цель состоит в том, чтобы назначить период производства для каждого заказа с учетом срока его выполнения и ограничений по мощности оборудования, чтобы минимизировать сумму затрат на хранение и затраты на переналадку.

Пример: Рассмотрим задачу со следующими входными данными: количество элементов nbItems=2; количество периодов nbPeriods=5; стоимость хранения h=2; время спроса на товары типа 1 d1t∈1,…,5=(0,1,0,0,1) и для предметов типа 2 d2t∈1,…,5=(1,0,0,0,1); g1,2=5, g2,1=3. Возможным решением этой проблемы является productionPlan=(2,1,2,0,1) что означает этот элемент 2 будет производиться в период 1; элемент 1 в период 2; элемент 2 в период 3 и элемент 1 в период 5. Обратите внимание, что в период 4 нет производства, это период простоя. Стоимость, связанная с этим решением, составляет g2,1+g1,2+g2,1+2∗h=15 но это не оптимальная стоимость. Оптимальное решение productionPlan=(2,1,0,1,2) со стоимостью g2,1+g1,2+h = 10.

**Задание 24.**

# 060: Совместное использование

Нужно редактировать постановку

**Задание 25.**

Проблема планирования проекта с ограниченными ресурсами

Запланирован ряд мероприятий. Каждое мероприятие имеет продолжительность и не может быть прервано. Существует набор отношений приоритета между парами мероприятий, которые гласят, что второе мероприятие должно начинаться после завершения первого. Набор отношений предшествования обычно задается в виде ориентированного ациклического графа (DAG), где ребро ( u , v ) представляет отношение предшествования, где u должно закончиться до начала v . DAG содержит два дополнительных действия продолжительностью 0, источник и приемник , где источником является первое действие, а приемник — это последнее действие (это фиктивные действия).

Есть множество возобновляемых ресурсов. Каждый ресурс имеет максимальную объем, и в любой заданный временной интервал может использоваться не более этого объема. Каждое мероприятие имеет спрос (возможно, нулевой) на каждый ресурс. У фиктивных мероприятий источника и приемника нулевая потребность во всех ресурсах.

Проблема обычно формулируется как проблема оптимизации, когда время выполнения (т. е. время завершения действия приемника ) сведено к минимуму.

## Многорежимный вариант

Расширением базового RCPSP является многорежимный вариант, в котором мероприятия могут иметь несколько режимов. Режим определяет продолжительность и потребности в ресурсах для мероприятия. В этом варианте расписание должно указывать режим каждого мероприятия, а также время его начала.

**Задание 26.**

Задача заключается в поиске оптимального компактного одиночного кругового расписания для спортивного мероприятия. Дан набор n команд, каждая команда должна сыграть против любой другой команды ровно один раз. В каждом раунде команда играет дома или на выезде, однако ни одна команда не может играть дома или на выезде более трех раз подряд. Сумма расстояний по передвижениям, каждой команды, должна быть минимизирована. Особенность этой проблемы заключается в заранее определенном месте проведения каждой игры, т. е. если команда a играет против b уже известно, состоится ли игра в доме a или в дома у b. Решение задачи считается сбалансированным, если количество домашних и выездных игр отличается не более чем на единицу для каждой команды; в противном случае оно называется несбалансированным или случайным. Несбалансированные решения могут быть невозможны.

Исходные расписания предполагают симметричные *круговые* расстояния: для i≥j, dij=dji=min(i−j,j−i+n).

Например, вот решение для некоторой задачи из 8 команд с круговыми расстояниями, в каждой строке которого описывается расписание для отдельной команды и указывается расстояние ее перемещения:

1 7 @4 @5 @6 2 @3 team travel distance: 14

@0 5 7 @6 @4 3 @2 team travel distance: 12

@3 4 6 7 @5 @0 1 team travel distance: 10

2 6 5 @4 @7 @1 0 team travel distance: 8

6 @2 0 3 1 @7 @5 team travel distance: 10

@7 @1 @3 0 2 @6 4 team travel distance: 10

@4 @3 @2 1 0 5 @7 team travel distance: 10

5 @0 @1 @2 3 4 6 team travel distance: 6

total cost: 80

Префикс «@» означает, что команда играет на выезде, например, команда 0 играет на выезде против команды 4 в третьем раунде.

**Задание 27.**

Проблема сбалансированной рабочей нагрузки медсестер

При заданном наборе больных, распределенных по ряду больничных зон, и имеющемся сестринском персонале необходимо закрепить за каждым больным по одной медсестре таким образом, чтобы работа распределялась между медсестрами поровну. Каждому пациенту присваивается приоритет, соответствующий объему ухода, который ему требуется; нагрузка медсестры определяется как сумма приоритетов пациентов, за которыми она ухаживает. Медсестра может работать только в одной зоне, и существуют ограничения как на количество пациентов, закрепленных за медсестрой, так и на соответствующую рабочую нагрузку. Мы балансируем рабочие нагрузки, минимизируя их стандартное отклонение.

Эту проблему можно разбить на две фазы: *укомплектование штатов медсестер* , которое распределяет медсестер по зонам, и *назначение медсестер-пациентов* , которое затем распределяет пациентов по медсестрам.

Вариант этой проблемы состоит в том, что пациенты сгруппированы в небольшое количество типов, и его острота связана с тем, что каждый тип пациента зависит от медсестры. Обратите внимание, что общая рабочая нагрузка больше не известна априори, так как состояние каждого пациента теперь зависит от того, к какой медсестре он приписан. Есть две цели, которые необходимо минимизировать: общая рабочая нагрузка и стандартное отклонение рабочих нагрузок.

**Задание 28.**

В колледже Святой Марии не работает электронная почта, и учитель хочет сообщить Роберту что-то срочное.

Учительница встречает Крейга и просит его передать Роберту, что она хочет поговорить с ним.

Крейг говорит, что если он встретит Роберта, все в порядке, но в противном случае он отправит сообщение всем, кого встретит, и сообщение пойдет дальше.

Каждый ученик говорит каждому встреченному ученику, что учительница ждет Роберта в своем кабинете.

Студенты встречаются друг с другом (мы не знаем, в каком порядке):

1) Крейг знакомится с Джоном и Джейсоном.

2) Джейсон встречает Кики, Адама и Дэвида.

3) Адам встречает Скотта и Джереми

4) Джереми встречает Джона и Скотта

5) Кики встречает Криса

6) Крис встречает Дэвида и Адама

7) Дэвид встречает Роберта

Думаете, учительница будет вечно ждать в своем кабинете?

Отобразите все возможные пути от Крейга до Роберта.

**Задание 29.**

Какое наименьшее число делится на 225 и состоит только из единиц и нулей?

Если начать с моего возраста в годах и применить четыре операции:

+2 /8

-3 \*7

в каком-то порядке, то окончательный ответ, который вы получите, — это возраст моего мужа в годах.

Как ни странно, если вы начнете с его возраста и примените те же четыре операции в другом порядке, то вы получите мой возраст.

Каковы наши два возраста?

Если начать с моего возраста в годах и применить четыре операции:

+2 /8

-3 \*7

в каком-то порядке, то окончательный ответ, который вы получите, — это возраст моего мужа в годах.

Как ни странно, если вы начнете с его возраста и примените те же четыре операции в другом порядке, то вы получите мой возраст.

Каковы наши два возраста?

**Задание 30.**

Джон вдвое старше Джима; сумма их возрастов такая же, как у Джерри, а возраст троих в сумме составляет 36 лет.

Сколько лет каждому?

**Задание 31.**

arch\_friends.mzn

Харриет, вернувшись из торгового центра, с радостью описывает свои четыре покупки обуви своей подруге Авроре. Авроре просто нравятся четыре разных вида туфель, которые купила Гарриет (эспадрильи цвета бежевого цвета, балетки цвета фуксии, фиолетовые туфли-лодочки и замшевые сандалии), но Гарриет не может вспомнить, в каком другом магазине (Foot Farm, Heels in a Handcart, The Shoe Palace, или Тутси) она получила каждую пару.

Можете ли вы помочь этим двоим выяснить, в каком порядке Гарриет купила каждую пару туфель и где она их купила?

1. Гарриет купила туфли цвета фуксии в магазине «Каблуки на тележке».

2. Магазин, который она посетила сразу после покупки своих фиолетовых туфель, был не Тутси.

3. Фут-Фарм был второй остановкой Харриет.

4. Через две остановки после выхода из «Обувного магазина» Харриет купила себе замшевые сандалии.

Определить: Заказ - Обувь - Магазин

**Задание 32.**

Обобщенная задача о назначениях — это задача минимизации затрат на назначение n сотрудников на m работ таким образом, чтобы каждый сотрудник был назначен ровно на одну работу.

Решите для матриц:

[|14, 5, 8, 7, 15,

| 2, 12, 6, 5, 3,

| 7, 8, 3, 9, 7,

| 2, 4, 6, 10, 1

|]

[|54, 54, 51, 53, 50,60,70,80,90,100,

|51, 57, 52, 52, 40,50,60,70,80, 90,

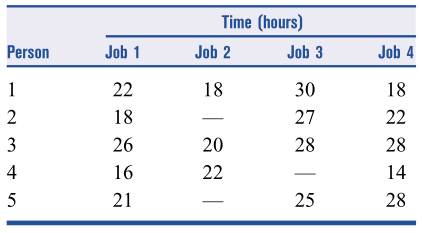
|50, 53, 54, 56, 40,50,60,80,93, 69,

|56, 54, 55, 53, 60,80,40,60,50,100

|]

**Задание 33.**

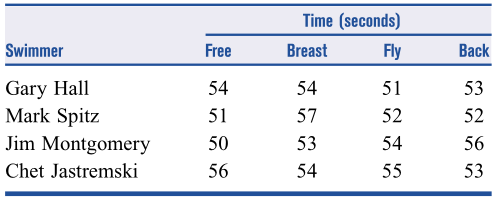
Пять сотрудников доступны для выполнения четырех работ. Время, которое требуется каждому человеку для выполнения каждой работы, указано в таблице. Определите распределение сотрудников по таким работам, при котором общее время, необходимое для выполнения четырех работ, будет минимальным. Каждый сотрудник назначается ровно на одну работу.



Примечание: прочерк означают, что человек не может выполнять эту конкретную работу.

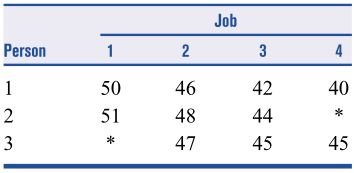
**Задание 34.**

Тренер собирает эстафетную команду для эстафеты на 400 метров. Каждый пловец должен проплыть 100 метров брассом, на спине, баттерфляем или вольным стилем. Тренер считает, что каждый пловец достигнет времени, указанного в таблице 51. Чтобы свести к минимуму время команды на гонку, какой пловец должен проплыть каким стилем?



**Задание 35.**

Компания принимает заявки на четыре строительные работы. Заявки на вакансии подали три человека. Их ставки (в тысячах долларов) приведены в Таблице (знак \* означает, что человек не делал ставки на данную работу). Человек 1 может выполнять только одну работу, а люди 2 и 3 могут выполнять до двух работ каждый. Определить минимальную стоимость закрепления лиц за рабочими местами.



**Задание 36.**

Решить задачу о назначениях.

[

13, 21, 20, 12, 8, 26, 22, 11,

12, 36, 25, 41, 40, 11, 4, 8,

35, 32, 13, 36, 26, 21, 13, 37,

34, 54, 7, 8, 12, 22, 11, 40,

21, 6, 45, 18, 24, 34, 12, 48,

42, 19, 39, 15, 14, 16, 28, 46,

16, 34, 38, 3, 34, 40, 22, 24,

26, 20, 5, 17, 45, 31, 37, 43]

**Задание 37.**

Мисс X, мисс Y и мисс Z, а также американка, англичанка и француженка, но не обязательно в таком порядке, сидели вокруг круглого стола и играли в черви. Каждая передала по три карты человеку справа от нее. Мисс Y передала три червы американке, Мисс X передала пиковую даму и две бубны тому, кто передал свои карты француженке.

Кем была американка? Англичанка? Француженка?

**Задание 38.**

Господин Бейкер, Дайер, Фармер, Гловер и Хозиер сидят за круглым столом и играют в покер. Каждый джентльмен является тезкой профессии одного из других.

Красильщик сидит двумя местами слева от мистера Хозиера.

Пекарь сидит на два места справа от мистера Бейкера.

Фермер сидит слева от мистера Фармера.

Мистер Дайер справа от перчаточника.

Как зовут красильщика?

**Задание 39.**

Каждый будний день Бонни присматривает за пятью соседскими детьми.

Детей зовут Кейт, Либби, Марго, Нора и Отто; фамилии Фелл, Гант, Холл, Айви и Джул. Каждому из них разное количество лет, от двух до шести. Можете ли вы найти полное имя и возраст каждого ребенка?

1. Одного ребенка зовут Либби Джул.

2. Кит на год старше ребенка Айви, который на год старше Норы.

3. Ребенок Фелла на три года старше Марго.

4. Отто вдвое больше, чем ребенку Холла.

Определить: Имя - Фамилия – Возраст

**Задание 40.**

Рассмотрим задачу о присвоении значений, нулей и единиц, позициям матрицы размера n x n, причем n четное, так, чтобы каждая строка и каждый столбец содержали ровно n/2 нулей и n/2 единиц.

Решите задачу для n = 4 и *n*=6.

**Задание 41.**

У вас есть пять тюков сена.

По какой-то причине их взвешивали не по отдельности, а во всех возможных комбинациях по два тюка. Вес каждой из этих комбинаций был записан и расположен в числовом порядке, без учета того, какой вес соответствовал какой паре тюков. Вес в килограммах составлял 80, 82, 83, 84, 85, 86, 87, 88, 90 и 91.

Сколько весит каждый тюк? Есть ли решение? Есть ли несколько возможных решений?

**Задание 42.**

На три доллара вы получаете 5 бананов, на пять долларов 7 апельсинов, на семь долларов 9 манго и на девять долларов три яблока, мне нужно купить 100 фруктов на 100 долларов. Пожалуйста, имейте в виду, что все виды фруктов должны быть куплены, но я не люблю бананы и яблоки, поэтому их должно быть минимальное количество.

**Задание 43.**

Моя банковская карта имеет 4-значный PIN-код, abcd. Я использую следующие факты, чтобы запомнить это:

- нет двух одинаковых цифр,

- двузначное число cd в 3 раза больше двузначного числа ab,

- двузначное число da в 2 раза больше двузначного числа bc.

Укажите PIN-код.

**Задание 44.**

Существует, ли биекция между {a, b, c, d} и { w, x, y, z}?

Если а не х, то с не у.

Если b равно y или z, то a равно x.

Если c не равно w, то b равно z.

Если d равно y, то b не равно x.

Если d не равно x, то b равно x.

**Задание 45.**

**best\_host**

Круглый обеденный стол с шестью местами для гостей. Некоторые гости, однако, не ладят друг с другом. Если два гостя, которые не ладят друг с другом, сидят рядом друг с другом, это вызовет конфликт за ужином. Как хозяин, вы должны расположить гостей в таком порядке, чтобы свести к минимуму конфликты.

Эндрю будет сидеть только рядом с Дейвом и Фрэнком;

Бетти будет сидеть только рядом с Карой и Эрикой;

Кара будет сидеть только рядом с Бетти и Фрэнком;

Дэйв будет сидеть только рядом с Эндрю и Эрикой;

Эрика будет сидеть только рядом с Бетти и Дэйвом;

Фрэнк будет сидеть только рядом с Эндрю и Карой.

На рис. показано следующее расположение:

Эндрю

Фрэнк Бетти

Эрика Кара

Дэйв

В приведенном выше примере рассадки есть три конфликта (Эндрю и Бетти, Кара и Дэйв, Эрика и Фрэнк).

Вопрос:

Какая рассадка сведет к минимуму количество конфликтов за ужином?

**Задание 46.**

INSTANCE: конечное множество U элементов, размер s(u) (положительное целое) для каждого u из U, вместимость b ­­– положительное целое число и положительное целое число k.

ВОПРОС: Существует ли такое разбиение U на непересекающиеся множества U1, U2, …, Uk, что сумма размеров элементов в каждом Ui не превосходит b?

data = 42 63 67 57 93 90 38 36 45 42

n = 10 // 10 numbers

m = 5 // 5 bins

b = 150 // bin capacity of 150

**Задание 47.**

Матрица NxN должна быть заполнена нулями и единицами так, чтобы:

- каждая строка и каждый столбец содержат равное количество нулей и единиц

- нет двух одинаковых строк или двух столбцов

- отсутствие последовательностей из 3 или более последовательных нулей или единиц в строках или столбцах

**Задание 48.**

В этом 2010 году моему старшему брату Андерсу будет столько же лет, сколько и мне в прошлом веке, а мне будет столько же лет, сколько ему в прошлом веке.

Мой брат старше меня на четыре года.

Сколько нам лет в этом году?

**Задание 49.**

2DPacking

Двумерная упаковка бункера

n прямоугольных предметов заданной высоты и ширины необходимо упаковать в прямоугольные ящики размером W x H. Предполагается, что предметы отсортированы по невозрастающей высоте.

N = 100;

W = 10;

H = 10;

ItemWidth = [ 2, 10, 9, 2, 6, 3, 3, 2, 10, 3, 3, 6, 6, 4, 3, 4, 6, 2, 3, 8, 9, 4, 7, 7, 2, 2, 7, 4, 6, 1, 1, 6, 7, 2, 8, 7, 4, 5, 4, 1, 5, 6, 5, 4, 3, 10, 3, 2, 9, 7, 4, 1, 1, 9, 4, 10, 2, 9, 4, 6, 2, 10, 6, 8, 3, 9, 1, 5, 4, 3, 2, 7, 7, 1, 8, 5, 4, 9, 6, 3, 7, 2, 1, 10, 8, 8, 7, 3, 3, 7, 10, 4, 1, 4, 10, 8, 2, 4, 6, 4 ];

ItemHeight = [ 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 8, 8, 8, 8, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 6, 6, 6, 6, 6, 6, 6, 5, 5, 5, 5, 5, 5, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1 ];