|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | |  |  |  | | МИНОБРНАУКИ РОССИИ | | | | Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |   Институт Информационных технологий | |
|  | |
| Кафедра Математического обеспечения и стандартизации информационных технологий | |
|  | |
|  | |

|  |  |
| --- | --- |
| **Практическая работа № 2** | |
| **по дисциплине** | |
| **«**Конфигурационное управление**»** | |
|  | |
| Выполнил студент группы | . |
| Принял преподаватель | . |

|  |  |  |
| --- | --- | --- |
| Лабораторная работа выполнена | «\_\_»\_\_\_\_\_\_\_202\_\_ г. | *(подпись студента)* |
|  |  |  |
| «Зачтено» | «\_\_»\_\_\_\_\_\_\_202\_\_ г. | *(подпись руководителя)* |

Москва 2023

**Оглавление**

[Задача 1 3](#_Toc148045239)

[Задача 3 4](#_Toc148045241)

[Задача 4 5](#_Toc148045242)

[Задача 5 6](#_Toc148045243)

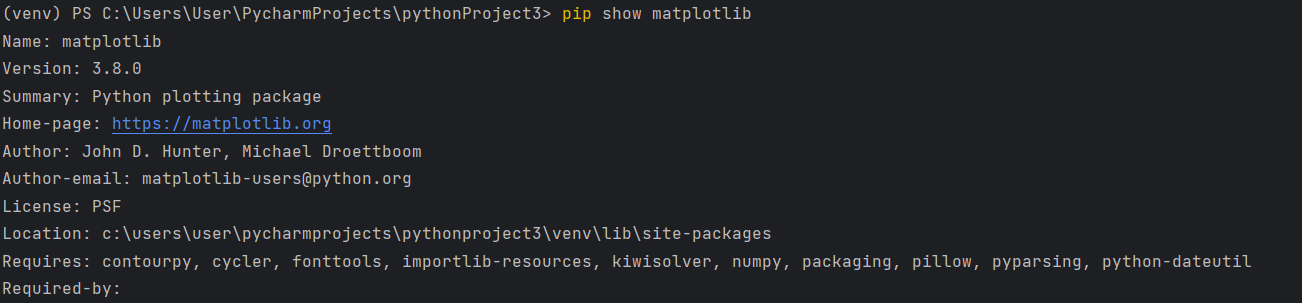
[Задача 6 7](#_Toc148045244)

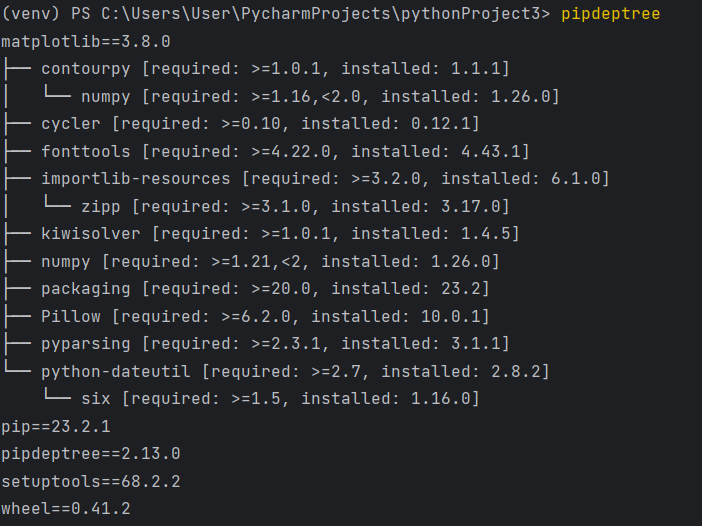
[Задача 7 8](#_Toc148045245)

# Задача 1

Вывести служебную информацию о пакете matplotlib (Python). Разобрать основные элементы содержимого файла со служебной информацией из пакета. Как получить пакет без менеджера пакетов, прямо из репозитория?

**Решение:**





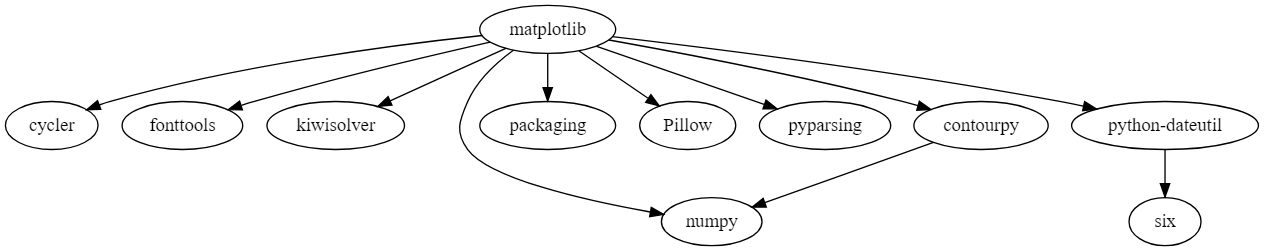


# Задача 3

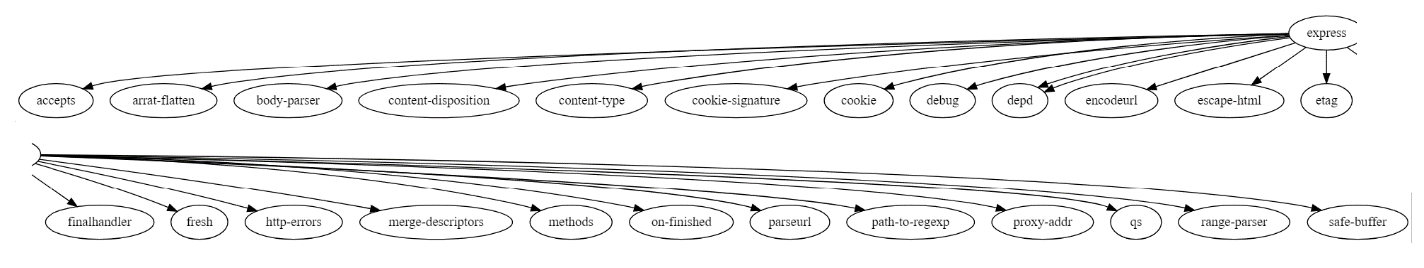
Сформировать graphviz-код и получить изображения зависимостей matplotlib и express.

**Решение:**

|  |
| --- |
| digraph G {  matplotlib->cycler;  matplotlib->fonttools;  matplotlib->kiwisolver;  matplotlib->numpy;  matplotlib->packaging;  matplotlib->Pillow;  matplotlib->pyparsing;    matplotlib->contourpy;  contourpy->numpy;    matplotlib->"python-dateutil";  "python-dateutil"->six;  } |



|  |
| --- |
| digraph G {  express->"accepts";  express->"arrat-flatten";  express->"body-parser";  express->"content-disposition";  express->"content-type";  express->"cookie-signature";  express->"cookie";  express->"debug";  express->"depd";  express->"depd";  express->"encodeurl";  express->"escape-html";  express->"etag";  express->"finalhandler";  express->"fresh";  express->"http-errors";  express->"merge-descriptors";  express->"methods";  express->"on-finished";  express->"parseurl";  express->"path-to-regexp";  express->"proxy-addr";  express->"qs";  express->"range-parser";  express->"safe-buffer";  } |



# Задача 4

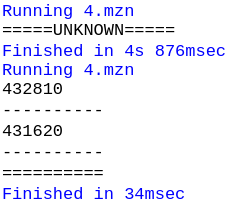
Изучить основы программирования в ограничениях. Установить MiniZinc, разобраться с основами его синтаксиса и работы в IDE.

Решить на MiniZinc задачу о счастливых билетах. Добавить ограничение на то, что все цифры билета должны быть различными (подсказка: используйте all\_different). Найти минимальное решение для суммы 3 цифр.

**Решение:**

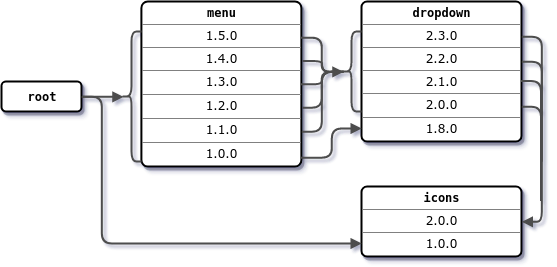
|  |
| --- |
| **include** "all\_different.mzn";  var **0**..**9**: a;  var **0**..**9**: b;  var **0**..**9**: c;  var **0**..**9**: d;  var **0**..**9**: e;  var **0**..**9**: f;  % ограничения  constraint all\_different([a,b,c,d,e,f]);  constraint a+b+c=d+e+f;  solve minimize d+e+f; %минимальное решение  output [show(a),show(b),show(c),show(d),show(e),show(f)]; |

**Результат:**



# Задача 5

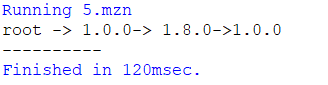
Решить на MiniZinc задачу о зависимостях пакетов для рисунка, приведенного ниже.



**Решение:**

|  |
| --- |
| var **0**..**5**:menu;  var **1**..**2**:dropdown1;  var **0**..**3**:dropdown2;  **var** int:**dropdown3**;  var **1**..**2**:icons;  constraint **if** menu=**0** then dropdown3=**8** /\ dropdown1=1 /\ icons=1 else dropdown1=2 /\ dropdown3=dropdown2 /\ icons=**2** endif;  solve satisfy;  output ["root -> 1.\(menu).0-> \(dropdown1).\(dropdown3).0->\(icons).0.0"] |

**Результат:**



# Задача 6

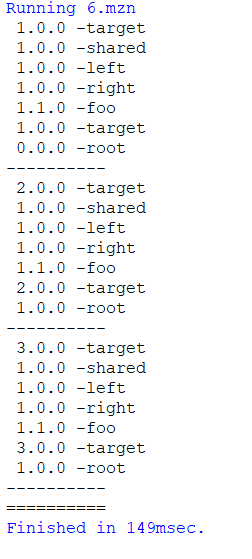
Решить на MiniZinc задачу о зависимостях пакетов для следующих данных:

|  |
| --- |
| root 1.0.0 зависит от foo ^1.0.0 и target ^2.0.0.  foo 1.1.0 зависит от left ^1.0.0 и right ^1.0.0.  foo 1.0.0 не имеет зависимостей.  left 1.0.0 зависит от shared >=1.0.0.  right 1.0.0 зависит от shared <2.0.0.  shared 2.0.0 не имеет зависимостей.  shared 1.0.0 зависит от target ^1.0.0.  target 2.0.0 и 1.0.0 не имеют зависимостей. |

**Решение:**

|  |
| --- |
| **include** "all\_different.mzn";  var **0**..**3**:root1;  var **0**..**9**:root2;  var **0**..**9**:root3;  var **1**..**3**:foo1;  var **0**..**9**:foo2;  var **0**..**9**:foo3;  var **1**..**3**:target1;  var **0**..**9**:target2;  var **0**..**9**:target3;  var **1**..**3**:left1;  var **0**..**9**:left2;  var **0**..**9**:left3;  var **0**..**3**:right1;  var **0**..**9**:right2;  var **0**..**9**:right3;  var **0**..**3**:shared1;  var **0**..**9**:shared2;  var **0**..**9**:shared3;  constraint **if** shared1 >= **1** /\ shared1 < 2 /\ shared2 <= 9 then left1 = 1 /\ right1 = **1** endif; %left **1.0**.**0** зависит от shared >=**1.0**.**0**., right **1.0**.**0** зависит от shared <**2.0**.**0**.  constraint **if** left1 >= **1** /\ right1 >= 1 /\ left1 < 2 /\ right1 < 2 then foo1 = 1 /\ foo2 = **1** endif; %foo **1.1**.**0** зависит от left ^**1.0**.**0** и right ^**1.0**.**0**., foo **1.0**.**0** не имеет зависимостей.  constraint **if** target1 >= **2** /\ foo1 >= 1 /\ foo1 < 2 /\ target1 <= **3** then root1 = **1** endif; %root **1.0**.**0** зависит от foo ^**1.0**.**0** и target ^**2.0**.**0**.  constraint **if** target1 >= **1** then shared1 = **1** endif; %shared **1.0**.**0** зависит от target ^**1.0**.**0**.  %foo **1.0**.**0** не имеет зависимостей и target **2.0**.**0** и **1.0**.**0** не имеют зависимостей..  solve maximize target1 + foo1;  output [" \(target1).\(target2).\(target3) -target \n \(shared1).\(shared2).\(shared3) -shared \n \(left1).\(left2).\(left3) -left \n \(right1).\(right2).\(right3) -right \n \(foo1).\(foo2).\(foo3) -foo \n \(target1).\(target2).\(target3) -target \n \(root1).\(root2).\(root3) -root \n"]; |

**Результат:**



# Задача 7

Представить на MiniZinc задачу о зависимостях пакетов в общей форме, чтобы конкретный экземпляр задачи описывался только своим набором данных.

**Решение:**

|  |
| --- |
| int: packs;  int: deps;  **set** of int: PACKAGES = **1**..packs;  array[PACKAGES] of **var** bool: **selected**;  constraint exists(selected);  array[**1**..**3**, **1**..**2**] of PACKAGES: dependencies;  constraint forall(i **in** **1**..deps)(selected[dependencies[i, **1**]] -> selected[dependencies[i, **2**]]);  **var** int: **numSelected** = sum(selected);  solve maximize numSelected;  output ["Is really installing packages: \(selected)"] |

