Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Кафедра інформатики

Звіт з Практичного завдання №1

По предмету: «Штучний інтелект для інформаційних технологій»

Виконав:

студент групи ІТІНФ-20-1

Самченко С. О.

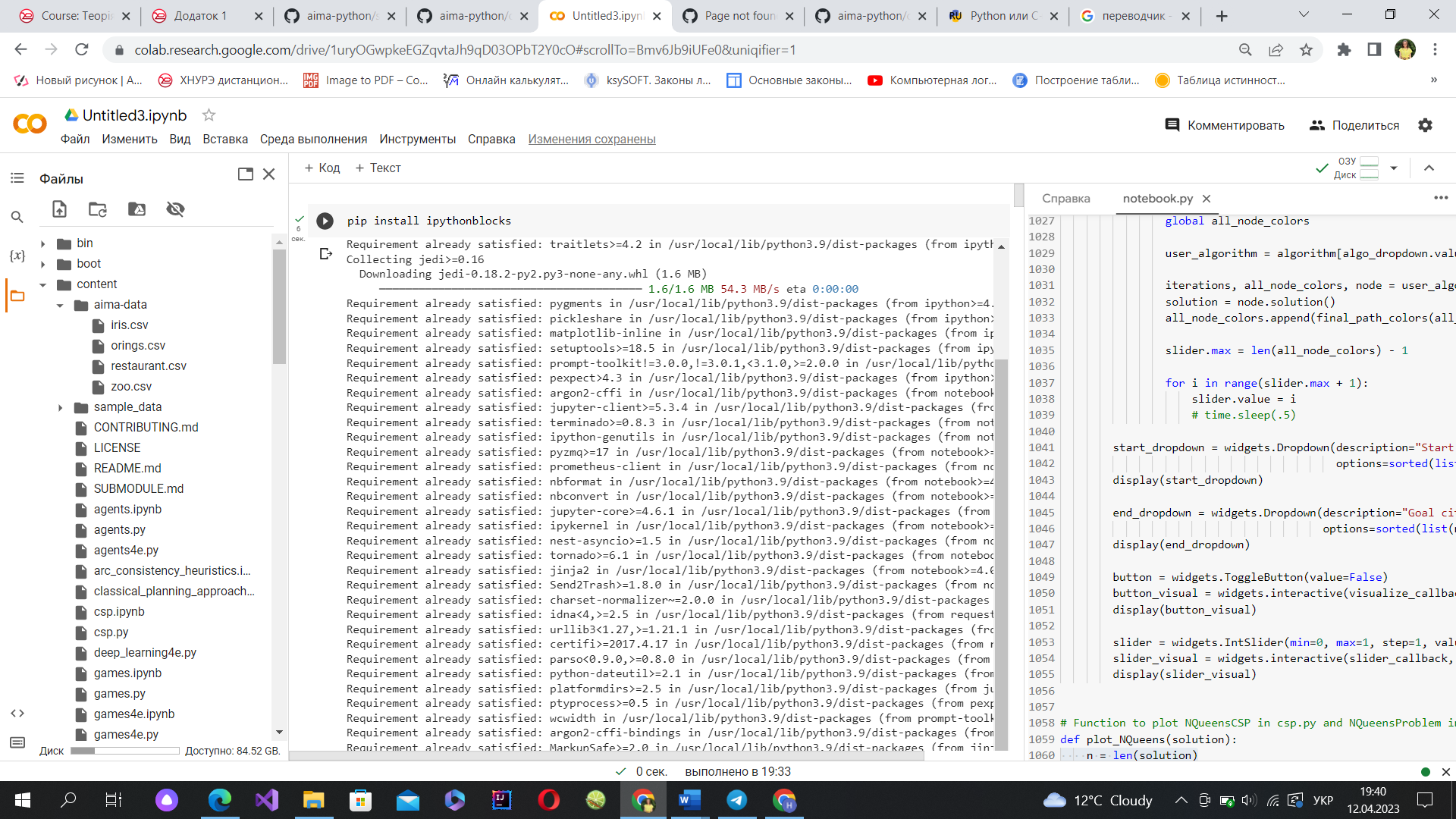
Харків 2023

**Мета**: ознайомитись з задачами виконання обмежень на практиці та з використанням мови Python.

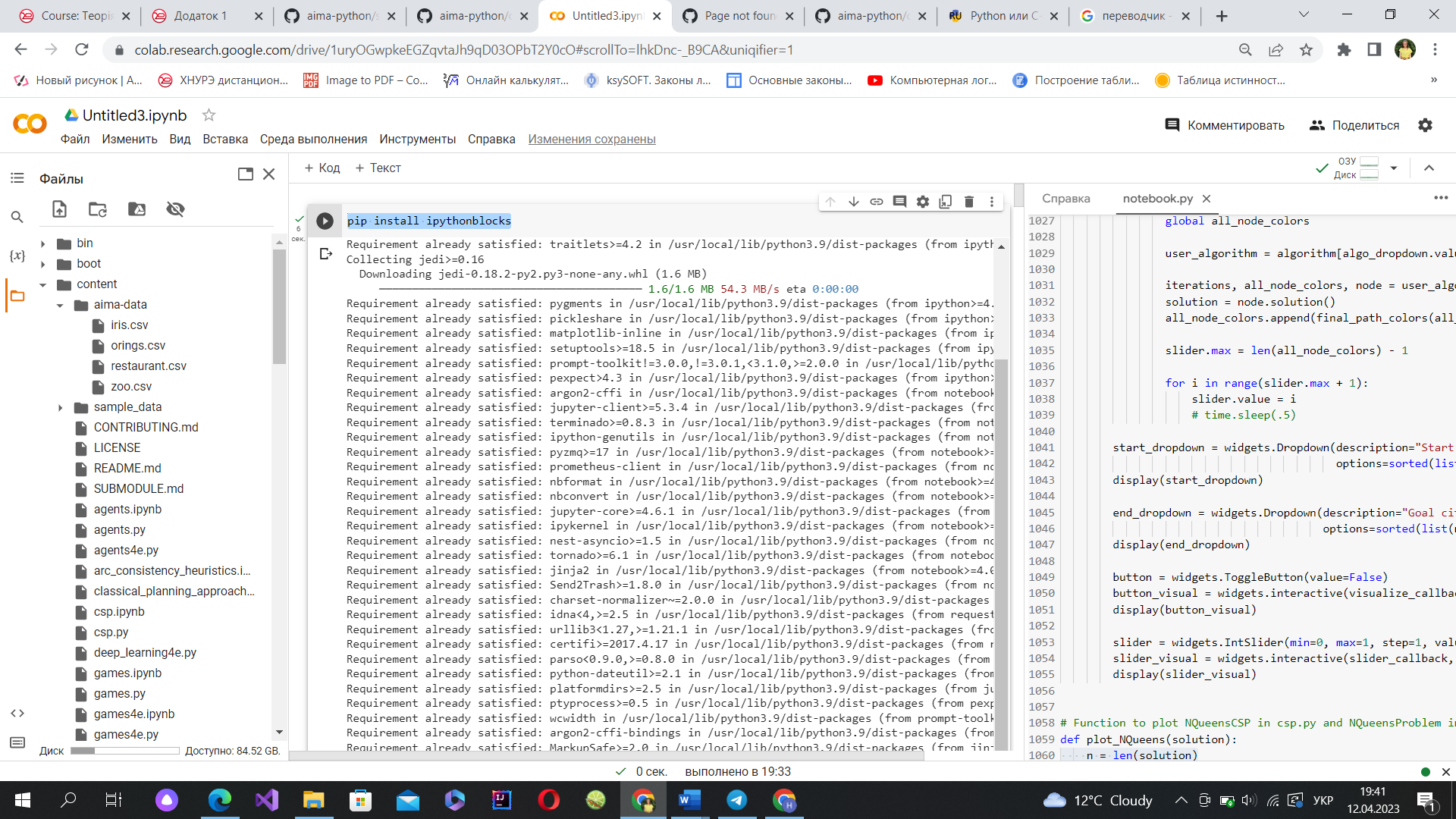
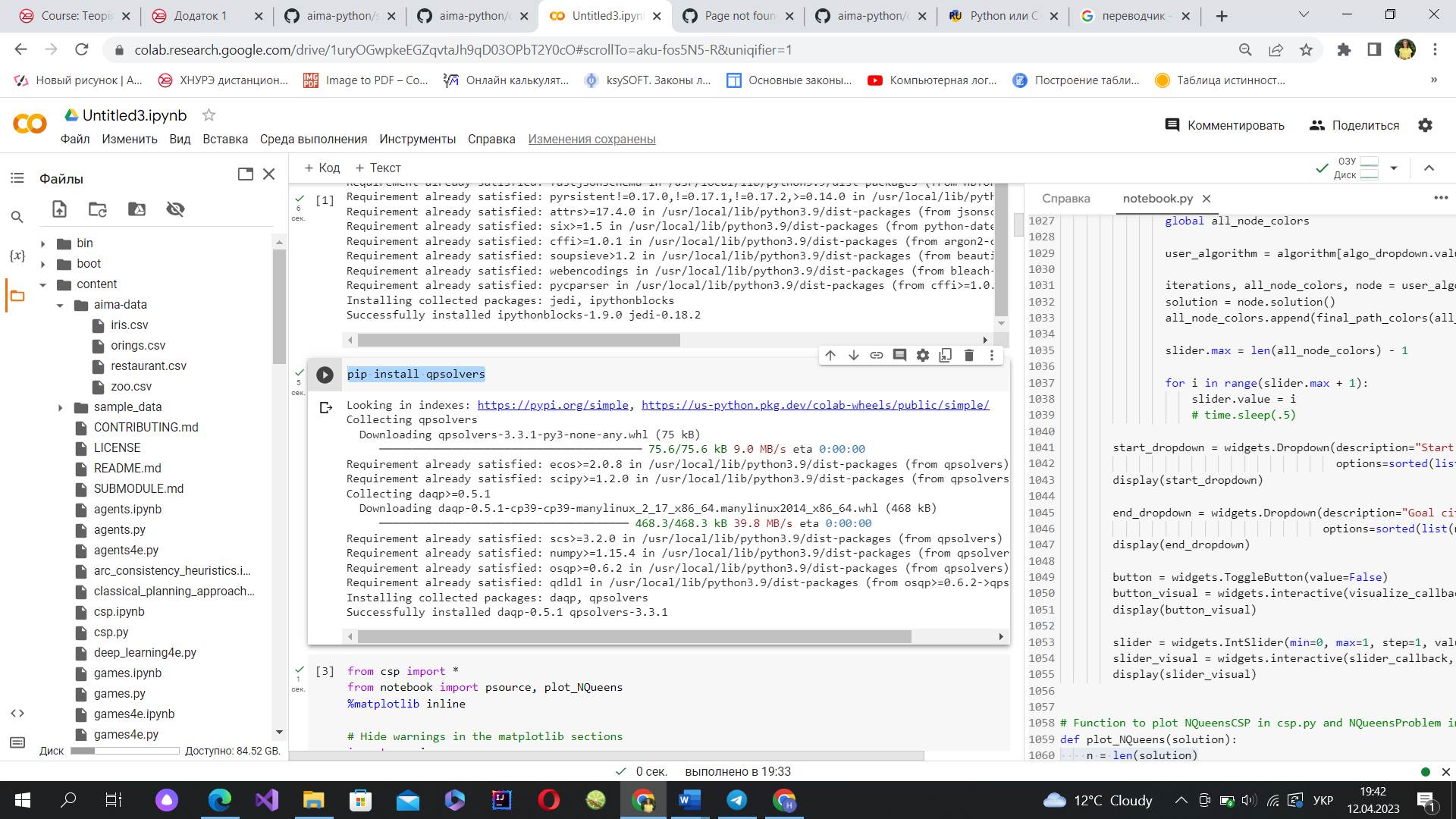
**Завдання:** Ознайомитись з реалізацією задачі виконання обмежень на мові Python https://github.com/aimacode/aima-python/blob/master/csp.ipynb. Візуалізувати роботу розв'язку задачі про **12 ферзів** за допомогою Step функції make\_plot\_board\_step\_function, реалізація якої наведена в посиланні вище.

**Хід роботи:**

1. Файли :



1. Імпорт залежностей:

1. Візуалізація роботи розв'язку задачі про 12 ферзів:

def label\_queen\_conflicts(assignment,grid):

    ''' Mark grid with queens that are under conflict. '''

    for col, row in assignment.items(): # check each queen for conflict

        conflicts = {temp\_col:temp\_row for temp\_col,temp\_row in assignment.items()

                         if (temp\_row == row and temp\_col != col)

                          or (temp\_row+temp\_col == row+col and temp\_col != col)

                          or (temp\_row-temp\_col == row-col and temp\_col != col)}

        # Place a 3 in positions where this is a conflict

        for col, row in conflicts.items():

                grid[col][row] = 3

    return grid

def make\_plot\_board\_step\_function(instru\_csp):

    '''ipywidgets interactive function supports

       single parameter as input. This function

       creates and return such a function by taking

       in input other parameters.

    '''

    n = len(instru\_csp.variables)

    def plot\_board\_step(iteration):

        ''' Add Queens to the Board.'''

        data = instru\_csp.assignment\_history[iteration]

        grid = [[(col+row+1)%2 for col in range(n)] for row in range(n)]

        grid = label\_queen\_conflicts(data, grid) # Update grid with conflict labels.

        # color map of fixed colors

        cmap = matplotlib.colors.ListedColormap(['white','lightsteelblue','red'])

        bounds=[0,1,2,3] # 0 for white 1 for black 2 onwards for conflict labels (red).

        norm = matplotlib.colors.BoundaryNorm(bounds, cmap.N)

        fig = plt.imshow(grid, interpolation='nearest', cmap = cmap,norm=norm)

        plt.axis('off')

        fig.axes.get\_xaxis().set\_visible(False)

        fig.axes.get\_yaxis().set\_visible(False)

        # Place the Queens Unicode Symbol

        for col, row in data.items():

            fig.axes.text(row, col, u"\u265B", va='center', ha='center', family='Dejavu Sans', fontsize=32)

        plt.show()

    return plot\_board\_step

twelve\_queens\_csp = NQueensCSP(12)

backtracking\_instru\_queen = make\_instru(twelve\_queens\_csp)

result = backtracking\_search(backtracking\_instru\_queen)

backtrack\_queen\_step = make\_plot\_board\_step\_function(backtracking\_instru\_queen) # Step Function for Widgets

matplotlib.rcParams['figure.figsize'] = (8.0, 8.0)

matplotlib.rcParams['font.family'].append(u'Dejavu Sans')

iteration\_slider = widgets.IntSlider(min=0, max=len(backtracking\_instru\_queen.assignment\_history)-1, step=0, value=0)

w=widgets.interactive(backtrack\_queen\_step,iteration=iteration\_slider)

display(w)

visualize\_callback = make\_visualize(iteration\_slider)

visualize\_button = widgets.ToggleButton(description = "Visualize", value = False)

time\_select = widgets.ToggleButtons(description='Extra Delay:',options=['0', '0.1', '0.2', '0.5', '0.7', '1.0'])

a = widgets.interactive(visualize\_callback, Visualize = visualize\_button, time\_step=time\_select)

display(a)

conflicts\_instru\_queen = make\_instru(twelve\_queens\_csp)

result = min\_conflicts(conflicts\_instru\_queen)

conflicts\_step = make\_plot\_board\_step\_function(conflicts\_instru\_queen)

iteration\_slider = widgets.IntSlider(min=0, max=len(conflicts\_instru\_queen.assignment\_history)-1, step=0, value=0)

w=widgets.interactive(conflicts\_step,iteration=iteration\_slider)

display(w)

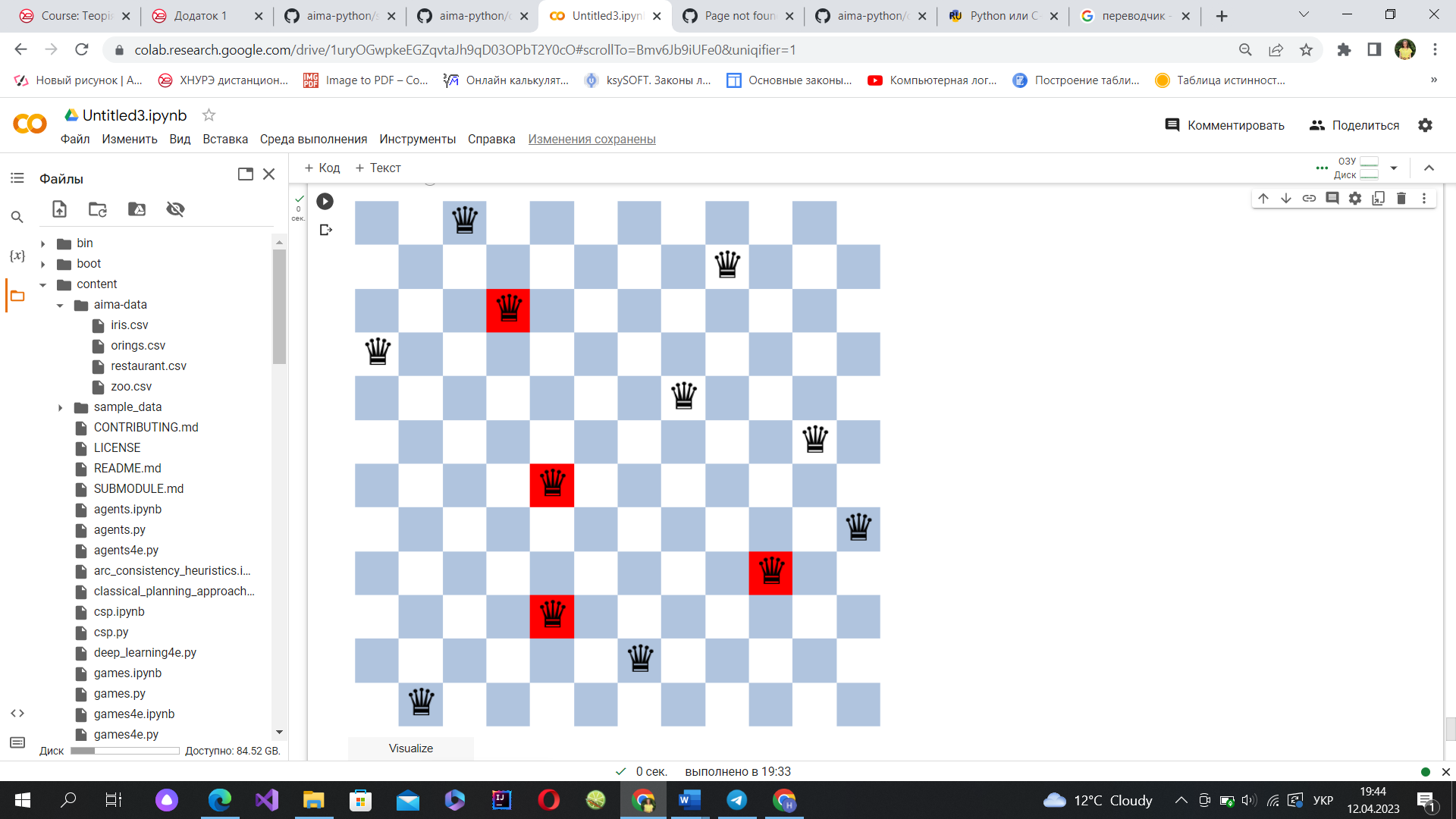
visualize\_callback = make\_visualize(iteration\_slider)

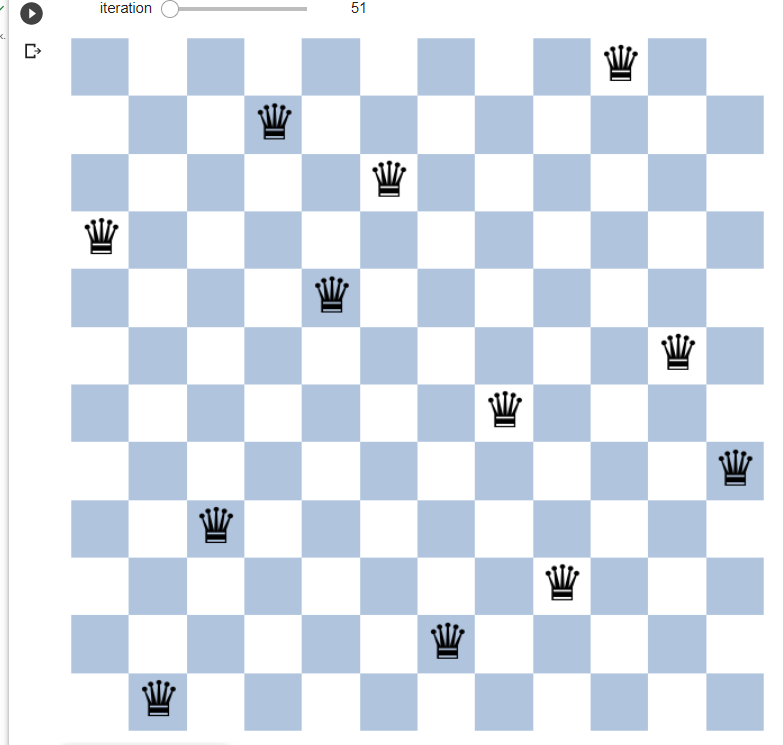
visualize\_button = widgets.ToggleButton(description = "Visualize", value = False)

time\_select = widgets.ToggleButtons(description='Extra Delay:',options=['0', '0.1', '0.2', '0.5', '0.7', '1.0'])

a = widgets.interactive(visualize\_callback, Visualize = visualize\_button, time\_step=time\_select)

display(a)





**Висновок:** у рамках даної роботи відбулося ознайомлення з задачами виконання обмежень на практиці та з використанням мови Python.