

## הוראות תפעול התוכנית

- נחלץ את הפרויקט מקובץ הzip.
- נתקין את כל החבילות הנדרשות מקובץ requirements.txt בעזרת פקוד הבאה:  
**pip install -r requirements.txt**

- על מנת להריץ את האלגוריתמים **CSP** ו-**WCSP** נריץ את הקובץ `SolvePureCSP.py`. נריץ אותו עם הפרמטרים הבאים:

1. סוג האלגוריתם
  2. סוג היוריסטיקה (עבור CSP בלבד)
  3. תאריך תחילת תקופת המבחנים
  4. תאריך סיום תקופת המבחנים
- היוריסטיקות האפשריות הן:

```
BACKTRACKING = backtracking
MINIMUM_REMAINING_VARS = mrv
DEGREE = d
LEAST_CONSTRAINING_VALUE = lcv
LEAST_CONSTRAINING_VALUE_AND_MINIMUM_REMAINING_VARS = combined
```

- לדוגמא, על מנת לפתור את הבעיה ע"י שימוש בCSP עם היוריסטיקה LCV נריץ משורש הפרויקט את הפקודה:

```
python SolvePureCSP.py csp lcv 2022/01/15 2022/03/08
```

לעומתו, על מנת לפתור את הבעיה ע"י שימוש בWCSP נריץ משורש הפרויקט את

הפקודה:

```
python SolvePureCSP.py wcsp 2022/01/15 2022/03/08
```

- ניתן לראות את התוצאות של האלגוריתם

- על מנת להריץ את האלגוריתמים **GA, GD, RGD, SA** נריץ את הקובץ `ISASolver.py`. נריץ אותו עם אותם פרמטרים כפי שמפורט לעיל מלבד היוריסטיקה.

לדוגמא, על מנת לפתור את הבעיה ע"י שימוש בGA נריץ את הפקודה:

```
python ISASolver.py ga 2022/01/15 2022/03/08
```

באופן דומה, על מנת לפתור את הבעיה ע"י שימוש בGD נריץ משורש הפרויקט את הפקודה:

```
python ISASolver.py gd 2022/01/15 2022/03/08
```

באותו אופן מריצים את הפקודות עבור `sa`, `rgd`.

עבור הרצה של `gai` או `gd` נוכל לבחור האם להרחיב את הבעיה או לא, לאחר סיום השיבוץ תופיע לנו

ההודעה הבאה:

```
Do you wish to assign halls as well? y/n
```

במידה ונרצה להרחיב ולשבץ גם אולמות את הבעיה נסמן y אחרת נסמן n.

- נוכל לראות מדדים להצלחת הפתרון של האלגוריתם בטרמינל, בנוסף ניתן לראות את שיבוץ המבחנים בGoogle Calendar בצורה מאוד נוחה ונגישה.  
לאחר השיבוץ בGoogle Calendar תופיע ההודעה הזו:

Do you want to save the calendar? Please insert y/n

לחיצה על y תשמור את השינויים בלוח ולחיצה על n תמחק אותם. אנו ממליצים למחוק את השינויים לאחר ההתבוננות כדי שלא יהיה צורך לעשות זאת ידנית שכן זהו תהליך מתיש.  
הגישה ליומן גוגל הינה בעזרת שם משתמש וסיסמה הבאים:  
User name: [aicalenderproject@gmail.com](mailto:aicalenderproject@gmail.com)  
Password: AIP12345!

כל שבועיים צריך לחדש את הרישיון ללוח, הגשנו את הקובץ המחודש, אך אם עברו יותר משבועיים ניתן להריץ את הקוד:

```
# scopes = ["https://www.googleapis.com/auth/calendar"]
# flow = InstalledAppFlow.from_client_secrets_file("../Utils/client_secret.json", scopes=scopes)
# credentials = flow.run_console()
# pickle.dump(credentials, open("../Utils/token.pkl", "wb"))
# credentials = pickle.load(open("../Utils/token.pkl", "rb"))
# service = build("calendar", "v3", credentials=credentials)
# result = service.calendarList().list().execute()
# export_to_calendar(courses, answer)
שנמצא מעל כל קריאה לcalendar בקובץ ISASolver.py בהערה.
```

- על מנת להריץ את הגרפים נריץ את הקובץ CreateGraphs.py נריץ אותו עם אותם הפרמטרים של הקובץ ISASolver.py  
למשל כדי לייצר את הגרפים עבור GA נריץ את הפקודה:  
**python CreateGraphs.py gd 2022/01/15 2022/03/08**

- על מנת להריץ את האלגוריתם ID3 נריץ את הקובץ ID3.py אשר נמצא בתיקיית ID3 בעזרת הפקודה:

**python ID3/ID3.py**

- נענה כן או לא על השאלות בהתאם. לדוגמא,

```
have you failed the exam? (yes, no)
yes
you should take the moed b exam
```

- בסוף נקבל עצה אם כדאי לגשת למועד ב או לא  
הערה: כשאר מריצים מהטרמינל הוא משום מה לא מדפיס את הצבעים:),  
על מנת לקבל את מלוא החוויה עם הצבעים יש להריץ דרך PyCharm 😊