**פרויקט בבינה מלאכותית  
"המומלצת המומלצת"**

תוכנית לימודים בהתאמה אישית

**מגישים:**

אפרת כהן – 315694166 - [cohen.efrat@campus.technion.ac.il](mailto:cohen.efrat@campus.technion.ac.il)

רותם סילפין – 312323835 – [silfinrotem@campus.technion.ac.il](file:///C:\Users\rotem\Downloads\silfinrotem@campus.technion.ac.il)

אביב ששון – 318305034 – [Avivsasson@campus@technion.ac.il](mailto:Avivsasson@campus@technion.ac.il)

**מנחה:**

דין צדוק - <deanzadok@cs.technion.ac.il>

**תאריך:** 20.3.2023A picture containing diagram

Description automatically generated

תוכן עניינים:

[הקדמה 3](#_Toc135548088)

[תיאור הבעיה והפתרון 4](#_Toc135548089)

[מבנה המערכת 5](#_Toc135548090)

[תרשים המערכת 6](#_Toc135548091)

[עוד מידע אודות המערכת 7](#_Toc135548092)

[גירוד המידע (Data Scraping) 7](#_Toc135548093)

[גרף הקורסים 8](#_Toc135548094)

[גרף האפשרויות 10](#_Toc135548095)

[אופטימיזציות בבניית גרף האפשרויות 11](#_Toc135548096)

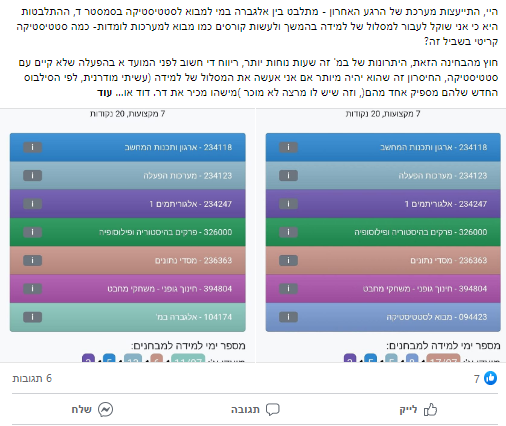
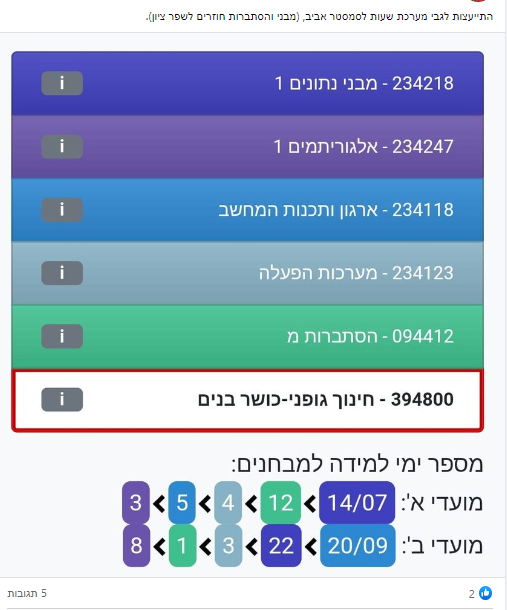
הקדמה

הטכניון – מכון טכנולוגי לישראל הוא אוניברסיטת מחקר ציבורית בחיפה המתמקדת בלימודי הנדסה ובמדעים מדויקים אך מלמדת גם רפואה ואדריכלות. הקמפוס הראשי של הטכניון ממוקם בנווה שאנן. בטכניון 18 פקולטות ויחידות אקדמיות. הטכניון הוא המוסד הוותיק בתחום המחקר הטכנולוגי בישראל ונחשב למוביל בתחום זה.

בטכניון הסטודנטים מחויבים לתוכנית הלימודים של המסלול אותו בחרו. חלק מהקורסים במסלול מוגדרים כקורסי חובה וחלקם ניתנים לבחירה חופשית של הסטודנט מתוך רשימה של קורסים אפשריים. סטודנט רשאי להרכיב לעצמו תוכנית לימודים סמסטריאלית כמעט ללא הגבלות, בתנאי שעומד בדרישות לקורסים אותם הוא מתכנן ללמוד.

הפקולטות מציעות תוכנית "מומלצת" שהיא למעשה המלצה לחלוקת הקורסים לפי סמסטרים על מנת שהסטודנטים יעמדו בדרישות השונות, תוך עמידה בתקופת הזמן שהוקצבה לתואר. הסטודנטים אינם חייבים ללמוד לפי סדר התוכנית המומלצת, ויכולים להשתמש בחלקה בלבד.

דעתם הרווחת של הסטודנטים היא שהמערכת המומלצת אינה "המומלצת ביותר" ולפעמים בהתאם לנסיבות (עומס, תרגילים רטובים, תרגילים יבשים וכו') כדאי לשנותה. בתקופת ההרשמה לקורסים קבוצת הפייסבוק של הפקולטות מתמלאות בפוסטים המבקשים המלצות על קומבינציות קורסים שונות עם הנסיבות השונות (עבודה, קושי גדול יותר מול רטובים וכו'). להלן דוגמאות:



מקור: קבוצת "מדעי המחשב טכניון" בפייסבוק.

# תיאור הבעיה והפתרון

**בעיה:**

רבים מהסטודנטים מאמינים שקיימת מערכת מומלצת טובה יותר עבורם מהמערכת המומלצת ע"י הפקולטה, דבר שמוביל אותם לבניה עצמאית של סילבוס הסמסטר. הסטודנטים נתקלים בקושי בבניית מערכת השעות הסמסטריאלית ותכנון תוכנית הלימודים עד לסוף התואר. הפתרון הקיים הוא ריבוי בהתייעצויות עם חברים לתואר, פוסטים רבים בקבוצת הפייסבוק של הפקולטה המבקשים את דעתם של סטודנטים אחרים ופתרונות נוספים.

פעמים רבות ניתן להרכיב מערכת "טובה יותר" עבור הסטודנט ולהקל בקושי במהלך הסמסטר, מערכת שלא בנויה נכון לסטודנט לפי העדפות, חוזקות והחולשות שלו יכולה לגרום לחוסר איזון בין תרגילי בית יבשים \ רטובים, קושי בקורסים ומבחנים, רמת עניין נמוכה ועוד וכל אלו יכולים להוביל לקושי רב. לעיתים קרובות אף עלול להוביל למצב של אי תקינות אקדמית, וכתוצאה מכך להביא לקושי נפשי, מנטלי, ואף פיזי. במקרה הטוב מצב של אי תקינות מביא למעקב של יועץ אקדמי ואזהרות חוזרות ונשנות בנוגע להמשך הלימודים של הסטודנט ובמקרה הגרוע הדחת הסטודנט ונשירה מהלימודים.

נוסף על כל אלו, קיימים קשיים נוספים בבניית מסלול אופטימלי לסיום התואר:  
מורכבות הדרישות במסלולים השונים - עולה קושי ב"ניתוב" ושיוך כל אחד מהקורסים השונים, דבר שמביא לקושי בהבנה בבניית מסלול נכון שיכול לסגור לסטודנט את התואר. למשל: סטודנט השלים מעל ל-18 נק"ז ברשימה א', עודף הנקודות "יעבור" לרשימה ב'. באופן דומה קורה עבור עודף בנק"ז עבור שרשרת מדעית.

חשיבות סדר לקיחת הקורסים בפקולטה – קיימים קורסים שהם קורסי קדם "בסיסיים" לקורסים רבים אחרים כך שיש להם חשיבות גדולה על מנת להתקדם, ולמנוע הארכה של התואר. בנוסף, קורסים אלה חשובים על מנת לאפשר לסטודנט ללמוד קורסי בחירה לאורך התואר. ישנם סטודנטים שאינם מודעים לחשיבות של סדר לקיחת הקורסים, דבר שלעיתים קרובות מוביל אותם להארכת התואר ולאי עמידה בקורסי קדם של קורסי בחירה ו"התפשרות" על קורסים שאינם מעניינים אותם, ולמעשה מצמצם את האפשרויות בחירה שלהם.

**פתרון:**

מערכת מבוססת בינה מלאכותית, שתמליץ לסטודנט על המסלול הטוב ביותר בהתאמה אישית עד לסיום התואר שיוביל להצלחה אופטימלית של הסטודנט בלימודים ולסיום התואר בהצלחה רבה.

# מבנה המערכת

**המערכת מורכבת מרכיבי הבסיס הבאים:**

1. Course – אוביקט של קורס בפקולטה, מכיל את שם הקורס, כמות נק"ז, דרישות קדם ולו"ז הרצאות, תרגולים ועוד.
2. State – מייצג את מצבו האקדמי של הסטודנט ביחס לתוכנית הלימודים.
3. Student – פרופיל סטודנט הקלט.

**בנוסף ישנם ארבעה חלקים עיקריים:**  
איסוף והכנת המידע כתשתית לכלל המערכת, גרף קורסי הפקולטה המאפשר ניהול דרישות הקדם של הקורסים, וגרף האפשרויות המייצג מסלולים חוקיים לסיום התואר. כך אנו יוצרים את התשתית לחלקה האחרון של המערכת - הרצת האלגוריתם A\* על מנת למצוא מסלול אופטימלי לסיום התואר.

1. **יצירת תשתית המידע Data –** אובייקט גלובלי, המרכז בתוכו את המידע הדרוש לאורך כל ריצת התוכנית.  
   פירסור קורסי הפקולטה מאתר הטכניון, יצירת כלל אובייקטי הקורסים ואחסונם במילון הקורסים - כאשר המפתח הוא מספר הקורס והערך הוא אובייקט הקורס. בנוסף נשמרים מסלול הלימודים של הסטודנט, החובות בתואר, סיווג הקורסים לפי דרישות וכו'.
2. **יצירת גרף הקורסים –** לאחר איסוף המידע, גרף הקורסים מייצג את כל הקורסים השונים במערכת והקשרים השונים ביניהם. זהו גרף מכוון חסר מעגלים (DAG): הצמתים בגרף הם הקורסים, וקשת בין קורס א' לקורס ב' מעידה שקורס א' הוא קדם או צמוד של קורס ב'. כאשר משמעותו של קדם היא שעל הסטודנט ללמוד את קורס א' לפני קורס ב', ושל צמוד שעל סטודנט ללמוד את קורס א' לפני או במקביל ללמידת קורס ב'. ניתן לתשאל את גרף הקורסים בעזרת הפונקציה get\_valid\_courses. פונקציה זו מקבלת כקלט רשימה של קורסים שהסטודנט השלים ומחזירה רשימה של קורסים שהסטודנט יכול ללמוד בסמסטר הקרוב, בהתאם לדרישות הקדם והצמוד. בעת בניית גרף האפשרויות פונקציה זו נקראת ומעדכנת אלו קורסים זמינים ליצירת כל האפשרויות השונות לסמסטרים חוקיים בהתאם לדרישות.
3. **יצירת גרף האפשרויות –** לאחר יצירת גרף הקורסים, נבנה גרף האפשרויות שבו כל צומת היא סמסטר ומסלול מייצג תוכנית חוקית לסיום התואר. הגרף נבנה בשלושה שלבים מרכזיים:
   1. בשלב הראשון, נעשה שימוש בפונקציה get\_valid\_courses השייכת לגרף הקורסים שכאמור מחזירה את רשימת הקורסים הזמינים לסטודנט בהתחשב בקורסים שכבר למד. הרשימה המתקבלת עוברת סינון להורדת קורסים לא רלוונטיים, לדוגמא: הורדת קורסים מדעיים אם הסטודנט כבר השלים שרשרת מדעית.
   2. בשלב השני, הרשימה נשלחת לפונקציה generate\_semesters שיוצרת את כל הסמסטרים האפשריים מרשימה זו בהתאם למגבלות השונות שהסטודנט בחר. הסמסטרים האפשריים שהתקבלו עוברים בדיקות שונות על מנת לוודא שהסמסטר אכן אפשרי מבחינת מערכת השעות, תאריכי מבחנים וכדומה.
   3. בשלב השלישי, הסמסטרים שעברו את כל הבדיקות מתווספים לגרף האפשרויות כצמתים, כל אחד מסמסטרים אלו מהווה שוב נקודת פתיחה כך שהחל ממנה יבוצעו שלושת השלבים הנ"ל באותו אופן, עד להגעה למצב שבו הסטודנט סיים את הדרישות של התואר.

בסיום יצירת הגרף, התקבלו כל המסלולים החוקיים לסטודנט החל ממצבו הנוכחי בלימודים ועד לסיום התואר.

1. **חיפוש –** לאחר שחושבו כל המסלולים החוקיים לסיום התואר עבור הסטודנט, אנו משתמשים באלגוריתם A\* עבור מציאת המסלול האופטימלי. לטובת החיפוש כתבנו יוריסטיקות המביאות לקבלת תוצאות שונות בהתאם להעדפות של אותה היוריסטיקה ועליהן נפרט בשלב הניסויים.

מימוש המערכת כולה נעשה בשפת python תוך שימוש בספריית networkx עבור הגרפים השונים ובספריות נוספות.

## תרשים המערכת

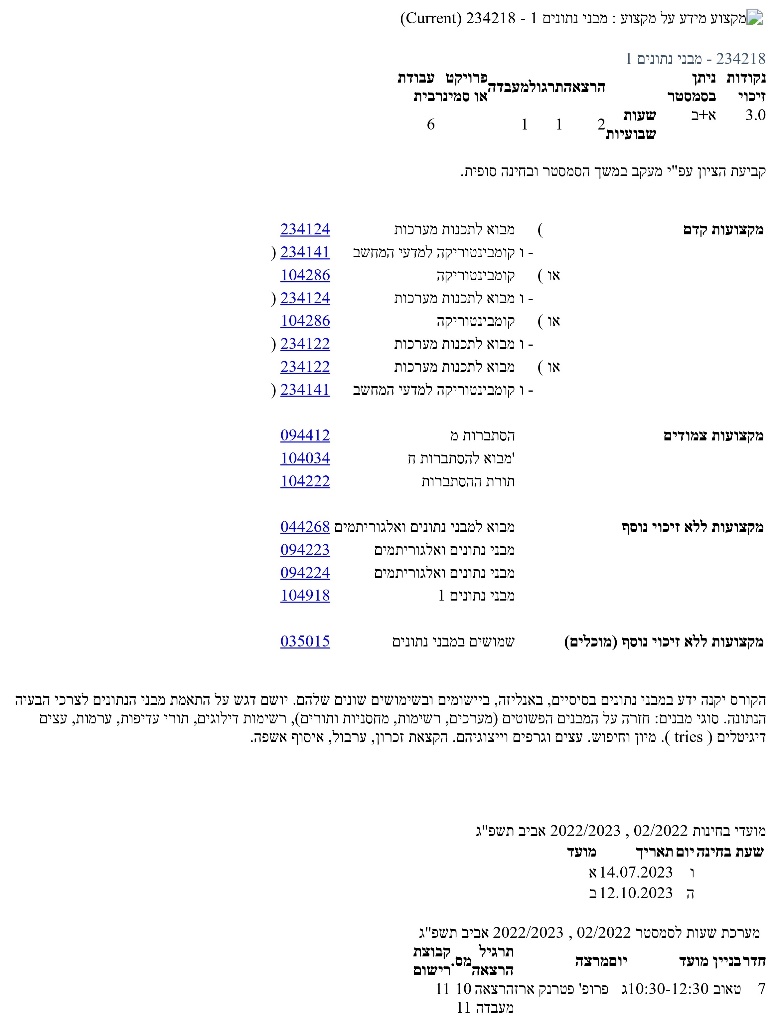
**תמונה שמכילה טקסט, צילום מסך, תרשים, עיצוב

התיאור נוצר באופן אוטומטי**

# עוד מידע אודות המערכת

## גירוד המידע (Data Scraping)

הוצאת המידע על הקורסים – על מנת להוציא את המידע על הקורסים בפקולטה, עלה צורך של פירסור הקטלוג. מחוסר זמן והיקף הפרויקט הוחלט להכניס באופן ידני כ-300 מספרי קורסים שונים המוצעים בקטלוג כחלק מקורסי החובה, רשימה א, וכו'. לאחר הוצאת כלל מספרי הקורסים, היה עלינו למשוך את המידע עליהם. החלטנו לפרסר את המידע מאתרי האינטרנט של הטכניון -   
" [https://www.graduate.technion.ac.il/Subjects.Heb/?SUB=<course\_id](https://www.graduate.technion.ac.il/Subjects.Heb/?SUB=%3ccourse_id)>".

לכל קורס, עמוד אינטרנט משלו. בעמוד האינטרנט מוצג כל המידע על הקורס למשל שם הקורס, מספר נק"ז, קורסי קדמים, לוח שעות ועוד. להלן דוגמה לעמוד הקורס מבני נתונים:

לפירסור המידע נעזרנו בספריית BeautifulSoup המאפשרת לנתח קוד HTML ו-XML בפייתון. הספרייה מאפשרת לגשת לדפי האינטרנט של הקורסים הרבים לצמצם את תוכנם, לחפש ולאסוף את המידע הרלוונטי. באמצעות הספרייה ניתן לבצע פעולות כמו שליפת תגי HTML, שליפת תכנים, חיפוש וסינון של מידע וכדומה. פעולת פירסור חיפוש ואיסוף המידע דרשו "עדינות" רבה מה שהיווה אתגרים רבים, ונקודות קצה רבות:

* עמודי הקורסים היו שונים במבנם מה שחייב זיהוי של המבנה והתאמת אלגוריתם הפירסור לפיו.
* מידע הקורס נשמר בשפה העברית העלה הקושי בקריאת המידע בצורתו הנכונה שכן הספרייה בה השתמשנו מותאמת לשפה האנגלית, לכן חלקי מידע שונים התערבבו מה שחייב סידור וכיוון של המידע מחדש.
* רבים מהקורסים הכילו אפשרויות רבות לקורסי קדם למשל כפי שניתן לראות בדוגמה המצורפת בקורס מבנה נתונים. על מנת לבנות את גרף הקורסים בחרנו בקורסי קדם שקיימים בקטלוג הפקולטה (אותם פרסרנו). במספר קטן של קורסים כלל דרישות הקדם לא היו מתוך קטלוג מדעי המחשב, לכן החלפנו את קורסי הקדם שלהם בקורסים שהם תחת "ללא זיכוי נוסף" או "מוכלים ללא זיכוי נוסף" שקיימים בקטלוג מדמ"ח.
* בשל מתקפת הסייבר על מערכות הטכניון עמודי האינטרנט נפלו, ולא היה ניתן להמשיך לעבוד איתם מהבית, לכן נאלצנו לעבור לעבודה עם קובץ json. לאחר זמן מה חזרנו לעבודה על הדפים Online על מנת לקבל את המידע המעודכן ביותר.

## גרף הקורסים

לאחר סיום שלב הParsing מתחילה בניית גרף הקורסים בעזרת הפונקציה create\_graph:

* לכל קורס נוצר צומת.
* לכל קורס במערכת שקיימות לו דרישות קדם מתווספת קשת עם mandatory=True.
* לכל קורס במערכת שקיים לו קורס צמוד מתווספת קשת עם linked=True.

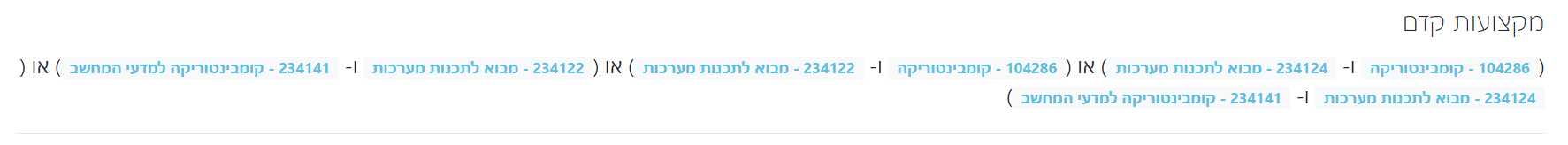
בתום בניית הגרף, אנו מבצעים הסרה של מעגלים מיותרים בגרף הקורסים בעזרת הפונקציה remove\_cycles, עליה נרחיב בהמשך, לאחר מכן הגרף הוא DAG ואנו מבצעים עליו מיון טופולוגי ושומרים אותו ברמת הclass.

לכל צומת קיימות 2 תכונות בוליאניות: Legacy, Dummy. בנוסף קיימים סוגי קשתות שונים לתיאור קשר של קדם או קשר של צמוד linked, mandatory.

לדוגמא - מכיוון שהקורס בהסתברות הוא צמוד של מבני נתונים, הקשת בינם מסוג linked.

לדוגמא – הקורס באינפי 1מ' הוא קדם לקורס באינפי 2מ' ולכן הקשת מסוג mandatory.

**התמודדות במקרה של כמה אפשרויות שונות לקורסי קדם של קורס:**

כשהתחלנו לעבוד עם הקורסים השונים במערכת, גילינו שלחלק מהקורסים אין רשימה בודדת של קורסי קדם, אלא כמה אפשרויות שונות. לדוגמא הקורס מבני נתונים. דרישות הקדם של הקורס:

דרישות הקדם כוללות כמה אפשרויות שונות שיש ללמוד אחת מבניהן טרם הסמסטר בו נלמד הקורס במבני נתונים.   
חלקן כוללות קורסים שהינם חלק מהקטלוג של מדעי המחשב תלת שנתי – על כן נמצאים אצלנו במערכת, וחלקן כוללות קורסים של מסלולים אחרים – מתמטיקה מדעי המחשב או המסלול בהנדסת מחשבים מהפקולטה להנדסת חשמל. בפועל רק דרישת קדם אחת רלוונטית לסטודנטים במסלול התלת שנתי. כדי לייצג קשר זה של קדמים בגרף הקורסים, קשר של "או", היה עלינו להציע פתרון מתאים.   
לאחר כמה ניסיונות להתמודד עם קורסים שמופיעים בדרישות הקדם/ הצמוד אך אינם מהמסלול התלת שנתי התגבשנו על הפתרון הבא:

* במידה וביצירת גרף הקורסים מופיע בדרישות הקדם/צמוד קורס שלא חלק מהקטלוג התלת שנתי, אנו מוסיפים קורס זה לגרף בתור צומת legacy. הוספה זו מאפשרת תמיכה בסטודנטים שעומדים בדרישות הקדם שאינן של המסלול התלת שנתי. יחד עם זאת, מבטיחה שנמשיך להציע לסטודנטים קורסים אך ורק מהמסלול התלת שנתי, מכיוון שבget\_valid\_courses אנו מסירים מהרשימה כל קורס שיש לו צומת מסוג legacy.
* בחירה של קדמים מהמסלול התלת שנתי : פעולה זו מבוצעת בשני שלבים. בשלב ההתחלתי באופן רוחבי בשלב איסוף המידע מתבצעת המרה של הקורסים שאינם מהמסלול התלת שנתי לקורס ללא זיכוי נוסף המתאים להם. בשלב הסופי בעת יצירת גרף הקורסים נעשה שימוש בselect\_cs\_pre\_courses שמבטיחה בחירה של אפשרות יחידה של קדמים – אלו שמכילים אך ורק קורסים מהמסלול התלת שנתי. בדוגמא של מבני נתונים תבחר האפשרות : מת"מ וקומבינטוריקה למדעי המחשב.

**מעגלים בגרף הקורסים:**

לאחר יצירת גרף הקורסים, הוא אינו DAG. קיימים קורסים לדוגמה: הקורס "פרויקט באינטרנט של הדברים" והקורס "אינטרנט של הדברים" שנדרשים ללמוד במקביל באותו הסמסטר. זוהי דרישה שונה מדרישת הצמוד שמאפשרת ללמוד לפני או במקביל.   
הקורס "פרויקט באינטרנט של הדברים" הוא צמוד של הקורס "אינטרנט של הדברים". לאחר יצירת הגרף אנו מאתרים מעגלים בגודל 2. כל מעגל בגודל 2 שכל קשתות המעגל הן מסוג צמוד, יהפוך לקורס אחד מאוחד שכולל את שני הקורסים – מכיוון שהקורסים חייבים להילמד באותו הסמסטר.

**Remove Cycles**

דוגמא לאיחוד הקורסים בהסרת מעגלים

לפני הצגת הפלט לסטודנט אנו מפצלים קורסים שהם איחוד ומציגים אותם כשני קורסים נפרדים בתוכנית הסמסטרים.  
לא נתקלנו במקרים כאלו, אך במידה וקיים מעגל שאינו "מעגל צמוד" ואורכו גדול מ-2 לא הגדרנו פתרון - הגרף אינו DAG ונזרקת שגיאה מתאימה. הנחנו זאת מכיוון שלא אמורים להיות מעגלים בגרף הקורסים, אם היו קיימים כאלו סטודנט לא היה יכול לסיים את התואר ולעמוד בדרישות קדמים / צמודים.

**מציאת הקורסים הזמינים לסטודנט בהינתן הקורסים שלמד עד כה:**

אנו עוברים על הצמתים לפי סדר המיון הטופולוגי. בהינתן שקורס טרם נלמד אנו נוסיף אותו לרשימת הקורסים הזמינים בתנאי שלא קיים מסלול בגרף בין קורס אחר שנוסף לרשימת הקורסים הזמינים לבין קורס זה. כך עבור קלט של קורסים שהסטודנט השלים עד כה בתואר, אנו מחזירים רשימה של קורסים שהסטודנט יכול לקחת.

## גרף האפשרויות

לאחר שיצירת גרף הקורסים מסתיימת, מתחילה בניית גרף האפשרויות ע"י הפונקציה create\_graph. הפונקציה מקבלת מהאובייקט הגלובלי Data את הState של הסטודנט ולוקחת את רשימת הקורסים שהסטודנט השלים. רשימה זו נשלחת לפונקציה get\_valid\_courses ששייכת לגרף הקורסים ומחזירה את רשימת הקורסים שהסטודנט רשאי ללמוד לפי הדרישות.  
רשימה זו עוברת מיון וסינון נוסף על מנת להגיע לזמן ריצה סביר והרחבה על כך מופיעה בהמשך.

בשלב הבא, רשימת הקורסים נשלחת לפונקציה generate\_semesters, פונקציה זו מורכבת מ2 חלקים:

1. קריאה לפונקציה רקורסיבית gen היוצרת את כל הסמסטרים האפשריים כך שגודלו המקסימלי של כל סמסטר הוא לפי המקסימום נק"ז שהסטודנט הכניס בהעדפות האישיות שלו. הסמסטרים שנוצרו נשמרים במילון זמני כאשר המפתח הוא כמות הנק"ז והערך הוא כל הסמסטרים שזהו סכום הנק"ז בהם.
2. בחלק השני, הפונקציה עוברת על כל הסמסטרים שסכום הנק"ז בהם הוא בטווח הנק"ז שהסטודנט בחר ובודקת האם הקומבינציה היא ויאבילית ע"י הבדיקות הבאות:
3. האם בסמסטר ישנם קורסים לא רלוונטיים?

לדוגמה: על הסטודנט להשלים קורס אחרון בשרשרת מדעית ולכן ברשימת הקורסים האפשריים היו 5 קורסים מדעיים שהסטודנט יכול לקחת אך אין צורך בסמסטרים שבהם יש יותר מקורס מדעי אחד.

1. האם הסמסטר עומד בדרישה של צמודים?

לדוגמה: הקורס הסתברות הוא צמוד(משמש כקדם אך ניתן ללמוד אותם כקורסים צמודים) של הקורס מבנה נתונים, כלומר סטודנט שצריך ללמוד מבנה נתונים חייב לעמוד בדרישת הצמוד. לכן בהכרח סמסטר חוקי הוא סמסטר שמכיל את מבני נתונים והסתברות יחד. ולכן סמסטרים המכילים את מבני נתונים בלבד אינם חוקיים(במידה ולא השלים הסתברות).

1. האם תאריכי המבחנים מסתדרים?

אחד מהנתונים שהסטודנט מזין למערכת הוא הפרש הימים שהוא רוצה בין מבחן למבחן ולכן כל סמסטר עובר בדיקה לוודא שהפרש הימים בין תאריכי המבחנים אכן תואם להעדפת הסטודנט.

1. האם קיים שיבוץ אפשרי למערכת השעות?

מתבצעת בדיקה האם ניתן לבנות מערכת שעות עבור הסמסטר, כלומר האם לכל קורס בסמסטר ניתן לשבץ בלוח הזמנים השבועי הרצאה/תרגול/ מעבדה וכו' לפי הדרישות של הקורס יחד עם שאר הקורסים בסמסטר. זוהי בדיקה שהתווספה כאופטימיזציה על מנת להקטין את רשימת הסמסטרים האפשריים ולכן הרחבה על בדיקה זו בהמשך.

הסמסטרים שעברו את הבדיקות בשלב השני אלו הסמסטרים החוקיים שהפונקציה מחזירה.  
יתכן ואף סמסטר לא יעבור את כל הבדיקות, ועל מנת להתמודד עם מקרה זה, גם הסמסטרים שסכום הנק"ז בהם היה נמוך מהטווח נשמרו קודם וכך ניתן לבדוק את הסמסטרים שסכום הנק"ז בהם הוא המקסימלי האפשרי מבין הסמסטרים שנשמרו. פעולה זו תתבצע עד שתתקבל רשימה לא ריקה של סמסטרים להחזיר.

לאחר קבלת רשימת הסמסטרים החוקיים, הפונקציה יוצרת גרף חדש – גרף האפשרויות עם צומת התחלה "start" ומוסיפה את הסמסטרים שהתקבלו כצמתים בגרף עם קשת בין הצומת start אליהם (שכבה ראשונה). סמסטרים אלו מייצגים את כלל האפשרויות לסמסטר שהסטודנט יכול ללמוד בסמסטר הבא ללימודיו.

לאחר יצירת "שכבת" הסמסטרים הראשונה, הפונקציה עוברת על כל אחד מסמסטרים אלו וקוראת לפונקציה: add\_optional\_semesters\_from\_current\_state שמקבלת סמסטר אפשרי וstate של הסטודנט. זוהי פונקציה רקורסיבית שצורת פעולתה כמעט זהה לcreate\_graph. הפונקציה מעדכנת את הstate של הסטודנט כך שלרשימת הקורסים שהוא השלים יתווספו הקורסים של הסמסטר שהתקבל כפרמטר. לאחר מכן גם כאן הפונקציה מקבלת את רשימת הקורסים האפשריים, הם עוברים סינון נוסף ונשלחים ל generate\_semestersעל מנת לקבל את כל הסמסטרים החוקיים. גם פה כל סמסטר שהתקבל מתווסף לגרף עם קשת בינו לבין הסמסטר שהתקבל כפרמטר. במעבר על הסמסטרים שמתקבלים, מתבצעת בדיקה האם זהו סמסטר שלאחר השלמתו הסטודנט מסיים את דרישות התואר ובמידה ואכן השלים את הדרישות הפונקציה לא תקרא שוב עבור סמסטר זה והוא מתווסף כעלה לגרף.

### אופטימיזציות בבניית גרף האפשרויות

במהלך הבנייה של גרף האפשרויות אחד מהקשיים העיקריים היה ההתמודדות עם יצירת כמות אקספוננציאלית של סמסטרים.  
בחלק מהמקרים התקבלו עשרות אלפי קומבינציות של סמסטרים מה שהותיר אותנו עם מאות סמסטרים חוקיים, כלומר עם מאות קשתות יוצאות בחלק מהצמתים בגרף. בנוסף חישוב הסמסטר הבא אחריו גם כן בוצע עבור כל אחד ממאות הסמסטרים הנ"ל ולכן זמן הריצה היה ארוך במיוחד.

בגרסה הראשונית של בניית הגרף, הבדיקות על כל סמסטר (קורסים רלוונטיים, צמודים, מבחנים וכו') בוצעו במהלך יצירת הסמסטר, דבר שהוביל לכך שהבדיקות בוצעו גם על סמסטרים שסכום הנק"ז בהם לא בטווח הרצוי. על מנת לחסוך בבדיקות עבור סמסטרים אלו, הפרדנו את יצירת הסמסטרים והבדיקות לשני חלקים שונים וכך הבדיקות בוצעו על כמות קטנה יותר של סמסטרים.

תיכנון מערכת שעות

פתרון נוסף שהעלינו הוא בדיקת היתכנות למערכת שעות. על מנת לצמצם את כמות הסמסטרים החוקיים הוספנו בדיקה המקבלת סמסטר ובודקת האם קיימת אפשרות לשיבוץ הדרישות השבועיות של כלל הקורסים ללא קונפליקטים. לכל קורס יתכנו ארבעה דרישות: הרצאה, תרגול, מעבדה ופרויקט. לא כל הדרישות רלוונטיות לכל קורס כלומר ישנם קורסים שיש בהם רק הרצאה או רק הרצאה ותרגול, בחלקם יש גם מעבדה ובחלקם יש רק פרויקט וכו'. עבור כל דרישה ישנה רשימת ימים ושעות שבהם מתקיים מפגש. לדוגמה: בקורס ספרתיות הדרישות השבועיות הן שלוש שעות הרצאה ושעתיים תרגול, עבור ההרצאה ישנן 2 אפשרויות: שעתיים רצופות בימי ראשון ושעה נוספת ביום רביעי, או שלוש שעות רצופות בימי שלישי. בנוסף ישנן 10 אפשרויות לזמני תרגול במהלך השבוע.

הקושי בבניית מערכת השעות נובע מכך שלעיתים מספר קורסים מתחרים על אותו חלון זמנים ועל הסטודנט לנסות לחזות איזו בחירה לא תיצור קונפליקטים ללא פתרון עם השעות של הקורסים האחרים. כלומר יתכן קורס א' עם הרצאה שבועית אחת בלבד שמתנגשת עם הזמנים של קורס ב' אך לקורס ב' קיימות שתי אפשרויות להרצאה שבועית ולכן על ההרצאה ללא ההתנגשות עם קורס א' להיבחר. פעולה זו נהיית מורכבת אף יותר כאשר בסמסטר ישנם 5-6 קורסים וכל בחירה שנעשית עלולה ליצור קונפליקט עתידי עם קורס אחר שיתכן כי עבור בחירה שונה הוא לא היה נוצר.

הפונקציה שבודקת זאת פועלת בצורה הבאה: ראשית מתבצעת בדיקה עבור אילו מהדרישות השבועיות אין קונפליקטים כלל, כלומר האם ניתן לשבצם בלוח הזמנים ואף דרישה של קורס אחר לא מתקיימת באותו זמן. לאחר בדיקה זו נותרנו עם הדרישות של כל קורס שבהם ישנן קונפליקטים במערכת השעות וצריכה להתבצע בחירה מושכלת. מכיוון שכל בחירה שנעשית משפיעה על הבחירות הבאות בחרנו להשתמש בשיטת brute force (אפשרות נוספת היא backtracking אך כיוון שסדרי הגודל הם לכל היותר עשר תרגולים, חמש הרצאות ושלוש מעבדות ההבדל ביניהם זניח). הפונקציה יוצרת עבור כל קורס רשימה שמחזיקה את כלל האפשרויות.

לדוגמה עבור הקורס ספרתיות שהוזכר לעיל, עם שתי אפשרויות להרצאה שבועית ועשר תרגולים יתקבלו עשרים קומבינציות - כל אחד מהתרגולים יחד עם כל אחת מההרצאות. לאחר מכן הפונקציה מנסה את כל הקומבינציות של כל קורס עם כל אחת מהקומבינציות של הקורסים האחרים.

במידה ונמצאה קומבינציה ללא קונפליקטים בלוח הזמנים, הסמסטר הוא חוקי. אחרת, הפונקציה ממשיכה לבדוק את כל האפשרויות ובמידה ולא נמצא אף שיבוץ של קומבינציות, הסמסטר מוסר מרשימת הסמסטרים הויאבילים.

צמצום רשימת הקורסים החוקיים

גורם נוסף שמשפיע על כמות הסמסטרים האפשריים זוהי רשימת הקורסים המתקבלת מהפונקציה get\_valid\_courses המחזירה את רשימת הקורסים שהסטודנט רשאי ללמוד בהתחשב בקורסים שכבר השלים. רשימה זו נשלחת לפונקציה שיוצרת את כל הסמסטרים האפשריים עם מגבלה יחידה - כמות מקסימלית של נק"ז בתואר, ולכן ככל שמספר הקורסים ברשימה גדול יותר, גם מספר הקומבינציות של סמסטרים גדל בהתאמה.

לדוגמה עבור סטודנט העומד בדרישות הקדם של 18 קורסים, מספר הקומבינציות לסמסטרים שניתן לבנות תחת מגבלת הנק"ז ומתוך רשימה זו הוא 69,792 סמסטרים אפשריים (אך לא בהכרח חוקיים). על מנת להתמודד עם הבעיה פעלנו בשתי דרכים כאשר הראשונה היא איחוד קורסים בעלי חשיבות זהה ויצירת קורסים כלליים, והשנייה היא הגבלת מספר הקורסים ע"י בחירה חכמה.

1. יצירת קורסים כלליים עבור בחירה חופשית, העשרה וספורט:

על סטודנט במסלול תלת שנתי במדעי המחשב לקחת 2 נק' של בחירה חופשית,6 נק' של קורסי העשרה(מל"ג) ו2 נק' של ספורט. קורסים אלו ללא קדמים או דרישות מיוחדות ולכן החלטנו להכניס למערכת שלושה קורסי מל"ג, קורס בחירה חופשית אחד, ושני קורסי ספורט כללים וזאת במטרה לצמצם את מספר הקורסים האפשריים. כך אנו מונעים את יצירת האפשרויות של אותו סמסטר אך עם קורס בחירה חופשית/ העשרה/ ספורט שונה בכל אפשרות.

לפי מגבלות לימודי הסמכה ניתן להירשם לכל היותר לקורס ספורט אחד בסמסטר לכן הכנסנו את הקורסים הכלליים, "ספורט 1" ו-"ספורט 2", כך שהקורס "ספורט 1" הוא קדם של הקורס "ספורט 2". כך אנו מונעים יצירה של סמסטר אפשרי עם 2 קורסי ספורט.

כיוון שלא ניתן ללמוד יותר מ2 קורסי מל"ג בסמסטר, אחד ברישום המוקדם ואחד נוסף ברישום המאוחר. אנו מגבילים יצירה של סמסטרים כאלו באופן הבא: למערכת הוספנו 3 קורסים "העשרה 1", "העשרה 2" ו-"העשרה 3" . כך שהקורס "העשרה 1" הוא קדם לקורס "העשרה 3" , דבר זה מאפשר למידה של לכל היותר 2 קורסי העשרה בסמסטר.

נתקלנו בקורסי סמינר רבים, ללא דרישות קדם עקב כך שאלו קורסים שהרישום אליהם נעשה באופן ידני ומשתנה לפי החלטת המרצה. רוב קורסי הסמינר דורשים ידע מעמיק במדעי המחשב והם פחות רלוונטיים לסטודנטים בסמסטרים הראשונים שלהם בתואר ולכן הוספנו תנאי כניסה מחמיר לקורסים אלו כך שהם יוצעו לסטודנט רק לאחר שהשלים 70 אחוז מקורסי החובה. ובאופן דומה לקורסי הבחירה, עם בקורסי סמינר קיימת הגבלה מטעם הפקולטה לסמינר אחד בסמסטר לכל היותר ולכן בוצע איחוד של כל קורסי הסמינר לקורס "מאוחד" אחד על מנת למנוע סמסטרים זהים למעט הסמינר וכך מקטינים את הקורסים האפשריים.

תנאי כניסה מחמיר נוסף שהוכנס הוא עבור קורסי "נושאים מתקדמים..", גם עבור רבים מקורסים אלו לא קיימים דרישות קדם(לעיתים הופיעו תחת הערות הקורס) אע"פ שאלו קורסים לסמסטרים מתקדמים יותר בתואר ולכן תנאי הכניסה שלהם הוא השלמת הקורס אלגוריתמים 1 – 234247.

1. הגבלת מספר הקורסים:

לאחר סינון הקורסים כמו שתואר בסעיף 1, קטן מספר הקורסים האפשריים בצורה משמעותית אך עדיין המספר הסופי היה גבוה מה שהוביל לזמן ריצה לא ריאלי. על מנת להתמודד עם הבעיה הוחלט להוסיף פרמטר חיצוני שניתן לשליטה בכל ריצה של התוכנית ומהווה חסם עליון למספר הקורסים האפשריים שמהם יבנו הסמסטרים.

בחירה שרירותית של כמות קורסים מתוך הרשימה בהתאם לפרמטר הביאה לבעיה חדשה: לדוגמא עבור סטודנט שנשאר לו להשלים קורס אחד מרשימה ב' וקורס ספורט אחד, היינו מצפים ליצירת סמסטר אחרון שבו יהיו קורס מרשימה ב' והקורס ספורט. ואכן התקבלה רשימה עם 18 קורסים – 17 מתוכם קורסי רשימה ב' וקורס אחד ספורט. בהנחה שהחסם על מספר הקורסים הוא 8, יתכן שע"י בחירה שרירותית יבחרו כל שמונת הקורסים מרשימה ב' ללא הקורס ספורט שדרוש מה שיוביל לבניית 2 סמסטרים, אחד המכיל את הקורס מרשימה ב' ואחד נוסף עם הקורס ספורט. לכן על מנת לפתור זאת, אנו מבצעים בחירה חכמה של הקורסים מתוך הרשימה.

ראשית, בוצע חישוב כמה נשאר באופן יחסי לכל סוג דרישה מתוך סך כל הדרישות שנשארו לסטודנט בתואר. לאחר מכן כמות הקורסים שנבחרת מכל דרישה תואמת לכמות הנק"ז היחסית שנשארה מדרישה זו. כך שעבור הסטודנט בדוגמא הנ"ל, שנותר לו 4 נק"ז לסיום התואר, 3 עבור הקורס מרשימה ב' ו1 עבור הקורס ספורט חושבו האחוזים באופן הבא: הקורסים מרשימה ב' הם 75% והקורס ספורט הוא 25% ולכן עבור פרמטר שמגביל ל8 קורסים יבחרו 7 קורסים מרשימה ב' והקורס ספורט.

כאשר נעשית בחירה של מספר מסוים של קורסים מתוך רשימה של קורסים גדולה יותר, הקורסים ממוינים לפי importance, ערך זה מחושב לכל קורס בגרף הקורסים ומייצג את החשיבות של הקורס לפי מספר הקורסים שהוא קדם שלהם. כך אנו דואגים שהבחירה היחסית של הקורסים לא תגרום לכך שקורסים שחשובים להמשך התואר לא מקבלים העדפה על פני קורסים אחרים שלא מהווים קדם לקורסים רבים אחרים. הגבלה זו נעשתה תוך מודעות לכך שהיא פוגעת בחישוב כלל האפשרויות, כלומר תתכן אפשרות לסידור הסמסטרים עד לסוף התואר שלא תופיע בגרף האפשרויות כי היא מורכבת מקורסים שלא נבחרו.

#### Pruning – גיזום סמסטרים

במהלך הבדיקות על גרף האפשרויות, הבחנו פעמים רבות ביצירת מסלולים ארוכים יותר ממסלולים שכבר קיימים בגרף האפשרויות, ככל הנראה מסלולים אלו לא יבחרו בריצת ה-A\* שכן אחת המטרות המרכזיות היא לסיים את התואר בזמן הקצר ביותר, לכן החלטנו לגזום מסלולים שניתן לקבוע שהם עתידיים להיות ארוכים יותר מהמסלול הקצר ביותר שנמצא עד כה, וכך קיצרנו את זמני הריצה.

במהלך יצירת המסלולים לסיום התואר, הגדרנו משתנה גלובלי בשם הshortest\_path\_length המייצג את אורך המסלול המינימלי שנמצא לסיום התואר עד כה. בכל הגעה של התוכנית לסמסטר שבו הסטודנט מסיים את התואר מתבצעת בדיקה האם אורכו של מסלול זה קטן יותר מהמסלול הקצר ביותר שידוע עד כה, וערכו של shortest\_path\_length מתעדכן בהתאם. בנוסף בתחילת הפיתוח של כל סמסטר, מתבצעת בדיקה שמנסה לחזות האם המסלול "כדאי". ניתן לקבוע זאת בהתחשב בכמות הנק"ז שנשאר לסטודנט לסיום התואר והאם יוכל לסיים את התואר תוך כמות סמסטרים שקטנה/ שווה למסלול המינימלי שידוע עד כה.

הבדיקה מתבצעת ע"י חלוקת כמות הנק"ז שנשארה לסטודנט לכמות הנק"ז המקסימלית שאפשרית בכל סמסטר (לפי בחירת הסטודנט) ובמידה ומספר הסמסטרים שנשארו יחד עם הסמסטרים שכבר תוכננו גדולים מאורך המסלול המינימלי מוותרים על פיתוח אותו צומת מתוך הנחה כי העדיפות של האלגוריתם A\* לאחר יצירת הגרף היא המסלול המהיר ביותר לסיום התואר תוך התחשבות בפרמטרים והעדפות הנוספות שהסטודנט הכניס למערכת.

#### שינוי "ותיעדוף" סמסטרים בפיתוח הגרף

כמה מהאופטימיזציות שהפעלנו נשענות על סכום הנק"ז בסמסטר. ע"י מיון הסמסטרים החוקיים לפי כמות הנק"ז ניתן להגיע לתוצאות טובות יותר שכן כך יתקבלו תחילה פיתוחים של סמסטרים עם כמות הנק"ז הגבוהה ביותר. אופן פיתוח זה יוביל לפיתוח המסלול הקצר ביותר לסיום התואר לפני מסלולים ארוכים יותר וכך יתבצעו גיזומים רבים יותר של סמסטרים (מסלולים) לא כדאיים.