מחלקת הנדסת חשמל

דו"ח פרוייקט מסכם- מבני נתונים ואלגוריתמים

מגישים: ניב קופולוביץ- 316352178

דני פורטנוב- 319397352

מוגש למרצה: עופר צור

תאריך הגשה: 23.07.2023

אופן ביצוע המטלה

בכדי לבצע את המטלה, יצרנו מספר מחלקות גדולות לכל ישות רלוונטית בפרויקט, כאשר לכל מחלקה יש משתנים ומתודות רלוונטיות לאותה מחלקה, הממשות את הפעולות שאנו מעוניינים לבצע.

מחלקת SyllabusDB

מחלקת SyllabusDB – מייצגת את הסילבוס, מסד הנתונים שמכיל בתוכו את כל הקורסים והמידע הרלוונטי עליהם.

תפקידה העיקרי לקרוא את הקובץ הcsv שניתן במטלה וליצור מכל קורס אובייקט מסוג Course או SpecialityCourse ולאחסן אותם.

בנוסף, המחלקה מאפשרת בהינתן קוד קורס לקבל את האובייקט של הקורס המתאים.

מבני נתונים בהם השתמשנו:

הקורסים מאוחסנים במבנה נתונים מסוג מילון (Dictionary). בחרנו להשתמש במבנה נתונים מסוג זה מאחר וה- Data Base המכיל את הנתונים אינו גדול (מספר הקורסים הכולל הוא בערך 90). חיפוש/הכנסה/הסרה ממילון מתאפשרים בסיבוכיות זמן ממוצע O(1) וסיבוכיות זמן במקרה הגרוע O(n). כאן, n=90 (ניתן לומר שהמספר אף יותר קטן, זאת מאחר ויש לנו מספר מילונים שכל אחד מכיל את הקורס המתאים לו (חובה/3 סוגי התמחויות) ולכן סיבוכיות הזמן במקרה הגרוע מתייחסת למספר קטן של נתונים (וידוע במקרה זה- מאגר קורסים לא גדול).

מחלקת SpecialityCoursesDB

מחלקת SpecialityCoursesDB – מאחסנת את כל קורסי ההתמחות של התמחות מסוימת.

ניתן להוסיף קורס או למצוא קורס בהינתן מספר קורס.

SyllabusDB משתמשת במחלקה זו על מנת לאחסן את קורסי ההתמחויות.

מבני נתונים בהם השתמשנו:

גם כאן בחרנו לאחסן את הקורסים במילון, מאותה הסיבה שה- Data Base אינו גדול באופן משמעותי.

<u>מחלקת Course</u>

מחלקת Course – מייצגת את הקורסים שקיימים בשנתון.

כל קורס מחזיק את פרטיו כפי שהם בסילבוס, בין היתר רשימת קורסי הקדם והקורס המקביל (אם ישנם).

בנוסף לכך, לכל קורס יש משתנה שאומר האם הקורס נלקח (הושלם כהלכה) או לא נלקח (התנאים ללקיחת הקורס לא מתקיימים).

המחלקה מכילה 2 מתודות עיקריות:

<u>validate_course</u>- מאמתת האם הפרטים של הקורס מקובץ הסטודנט תואמים לפרטים במסד הנתונים:

1. במידה וקוד הקורס לא תואם - פרטיו נשמרים ומצורפת אליהם הודעה מתאימה המסבירה על השגיאה. ההודעה מודפסת בסוף בקובץ הפלט.

2. במידה וקוד הקורס תואם אך יתר הפרטים לא תואמים (שם הקורס ומספר נק"ז), הקורס כן מאומת אך מצורפת הודעה המסבירה על השגיאה בסוף הקובץ.

3. במידה וכל הפרטים תואמים לא מצורפת אף הודעה.

<u>is_finished_properly</u>- בודקת האם בוצעו קורסי הקדם והקורס המקביל של קורס מסוים.

במידה ולפחות אחד מקורסי הקדם לא נלקח, הפונקציה מחזירה False ומצורפת הודעה מתאימה אודות הקורס החסר.

בנוסף החלטנו שברגע שקורס קדם לא בוצע, הקורס "נפסל" ומעדכנים את המשתנה שאומר האם הקורס נלקח או לא ל- "לא נלקח". כך למעשה כל קורס שתלוי בקורס שנפסל, ייפסל גם משום שקורס הבסיס לא נלקח.

מבני נתונים בהם השתמשנו:

את קורסי הקדם מאחסנים ברשימה- כאשר נעבור על קורסי הקדם של קורס מסוים נעבור על 4 קורסים לכל היותר- מספר קטן וקבוע של קורסים ולכן סיבוכיות הזמן היא O(n) במקרה הגרוע כאשר n הוא מספר קורסי הקדם.

מחלקת SpecialityCourse

מחלקת SpecialityCourse – יורשת ממחלקת Course.

מכילה מילון שהKey שלו הוא שם התמחות, והValue הוא סוג הקורס בהתמחות הזאת (חובה, בחירה או לא קיים).

המחלקה מכילה את המתודות:

get_speciality_course_type- מחזיקה, בהינתן התמחות, את סוג הקורס בהתמחות.

למשל בקורס 'עיבוד תמונה': עבור 'מחשבים' המתודה תחזיר 'בחירה', עבור 'אותות' תחזיר 'חובה' ועבור 'התקנים' תחזיר 'לא קיים'.

check_if_hw_sw- מחזירה את סוג קורס החובה בהתמחות מחשבים, תוכנה או חומרה.

במידה ואינו קורס חובה בהתמחות מחשבים הפונקציה לא תחזיר כלום.

מבני נתונים בהם השתמשנו:

בחרנו להשתמש במילון בעל 3 מפתחות (מפתח לכל סוג התמחות- מספר קבוע של מפתחות).

סיבוכיות הזמן במקרה הגרוע תהיה O(n) עבור חיפוש/הכנסה במילון, כאשר n הוא מספר ההתמחויות.

מחלקת Constants

מחלקת Constants – מחזיקה קבועים שונים: הודעות שגיאה, מילים קבועות, Enums ומילון שמכיל את הקודים של קורסי פרויקט הגמר.

הקבועים המוגדרים במחלקה:

Internships- מכיל את השמות של 3 הסוגים של פרויקטי הגמר שניתן לעשות.

Speciality- מכיל את השמות של 3 ההתמחויות שקיימות.

CourseType- מכיל את הסוגים שקורס יכול להיות; חובה, קורס בהתמחות ראשית או משנית, ובחירה מחוץ להתמחות.

SpecialityCourseType- מכיל את הסוגים של קורס התמחות; האם הוא בחירה, חובה או לא קיים בהתמחות.

ComputersCourseType- מכיל את הסוגים של קורסי החובה בהתמחות מחשבים; חומרה או תוכנה. בנוסף מכיל גם שדה TOTAL שמשתמש לייצוג של כמות קורסי החובה שנלקחו.

מחלקת Student

מחלקת Student – מייצגת את הסטודנט שעבורו נבדקת הזכאות ע"י המערכת.

מכילה מספר מבני נתונים:

spc_must_courses_req- מילון המאחסן את כמות קורסי החובה שצריך בהתמחות הראשית והמשנית. במידה ואחת ההתמחויות היא מחשבים, לקורסי החומרה והתוכנה תישמר הכמות בנפרד, כמו גם כן הסך הכל.

credits_taken- מילון שסופר את כמות הנק"ז שנלקחו בכל הסוגים: חובה, התמחויות ובחירה מחוץ להתמחות.

spc_must_courses_taken- מילון שסופר את כמות קורסי החובה שנלקחו בפועל בכל התמחות.

speciality_courses ו- mandatory_courses- מילונים ששומרים את קורסי הבחירה והחובה (בהתאמה) שנלקחו.

*הערה: במחלקת Student בחרנו להשתמש במילונים על מנת לאחסן את המידע מאותה סיבה שגודל ה-Data Base אינו גדל ועל כן מספר הקורסים שהסטודנט יכול לרשום בקובץ מוגבל. סיבוכיות חיפוש/הכנסה/הוצאה ממוצעת: O(1), במקרה הגרוע: O(n) כאשר n לא גדול בצורה משמעותית (גודל ה- DB ידוע לנו).

המחלקה מכילה את המתודות:

read_student_data- זו המתודה הראשונה שמבוצעת כאשר יוצרים סטודנט.

תפקידה לקרוא את הקובץ, ולבדוק את שמות ההתמחות הראשית או המשנית. במידה ואחד מבין שמות ההתמחות אינו תקין, מתווספת שגיאה בקובץ הפלט ובדיקת הקורסים לא תתבצע.

כאשר עוברים על רשימת הקורסים שהסטודנט ביצע, מאמתים כי הפרטים של כל הקורסים נכונים:

1. במידה וקוד הקורס אינו נכון, הפרטים נשמרים במילון לשם הדפסה.
2. במידה וקוד הקורס תקין הוא מתווסף למבני הנתונים הרלוונטיים ונבדקים יתר הפרטים שהם אופציונליים (אם אין התאמה בהם אז מודפסת הודעה אך הקורס כן נחשב).

generate_result_file- קוראת למתודה- run_courses_check ואחריה יוצרת את קובץ הפלט.

בקובץ הפלט רשום האם הסטודנט זכאי או לא זכאי לתואר, סכום הנקודות הכולל שנלקח, סכום נקודות קורסי ההתמחויות, פרטיו של הסטודנט, סוג פרויקט הגמר (אם נלקחו 2 הקורסים) והודעות/שגיאות אם קיימות.

run_courses_check- מגיעים למתודה זו רק אם שמות ההתמחויות הראשית והמשנית נקלטו ואמותו לפי השם הדרוש.

אופן הפעולה של המתודה:

1. ראשית בודקים עבור כל הקורסים, לפי הסדר בו הם רשומים בקובץ, האם הם הושלמו כהלכה. במידה ולא, מסירים אותם ממבני הנתונים הרלוונטיים ושומרים את ההודעות.
2. לאחר מכן בודקים ומחשבים את מספרי הנקודות של כל קורסי החובה ומוודאים את כמות הנק"ז של הקורסים הכלליים והספורט.
3. לאחר שעברנו על קורסי החובה, יש בידינו את המידע לגבי סוג פרוייקט הגמר שלקח הסטודנט (תעשייה/מחקר/פרויקט גמר במכללה) וכעת נעדכן את סוג פרויקט הגמר שנלקח.
   על מנת לומר שסטודנט בחר סוג מסוים של פרויקט גמר, עליו לבחור את 2 מספרי הקורס המתאימים לסוג הפרויקט שבחר (למשל לבחור את מספרי הקורס התואמים את סוג הפרויקט באופן הבא: תעשיה א+תעשיה ב, ולא לבחור מספרי הקורס שלא מתאימים לאותו סוג פרויקט לדוגמא: תעשיה א+מחקר ב).
   3.1. במידה ולא נמצאו 2 מספרי הקורסים של פרויקט גמר מסוג מסוים לא נבדוק את קורסי ההתמחות ושומרים הודעה מתאימה.
   3.2. במידה ונמצאו מספר התאמות לסוג המסוים של פרויקט גמר (למשל סטודנט כתב בקובץ שלו שביצע את תכן הנדסי בתעשיה ובנוסף ביצע גם מחקר), לא נבדוק את קורסי ההתמחות ושומרים הודעה מתאימה.
   3.3. במידה ונמצאה התאמה ל2 מספרי קורס של פרויקט גמר מסוג מסוים (ורק לסוג אחד כזה!), נעדכן את סוג הפרויקט שנלקח.

לאחר שעידכנו את סוג הפרויקט, נמשיך לבדוק ולחשב את כמות הנקודות של קורסי ההתמחות:

*הערה: אם פרויקט הגמר הוא בתעשייה, אנחנו לא נבדוק את ההתמחות המשנית באף אחד מהבדיקות הבאות.

עלינו לעבור על קורסי ההתמחות שביצע הסטודנט ולבדוק האם הגיע לכמות הנקודות הנדרשת בהתמחות ראשית, משנית (אם רלוונטי) ובבחירה מחוץ להתמחות.

הרעיון עליו התבססנו הוא לבדוק את כל הקומבינציות של קורס ובהתאם לסווג אותו להתמחות שבו הוא צריך להיות (כל בדיקה מתבצעת על כל סוג התמחות ראשית ומשנית):

1. מוסיפים את כל קורסי החובה שקיימים רק בהתמחות מסוימת, לחישוב הנקודות של אותה התמחות.
2. בודקים את כל קורסי החובה בהתמחות מסוימת שהם בחירה בהתמחות השניה:
   2.1. במידה ואין מספיק קורסי חובה שנלקחו, מוסיפים אותם לחישוב הנקודות של ההתמחות שבה הקורסים נחשבים קורסי חובה.
   2.2. במידה ויש מספיק קורסי חובה, שומרים אותם בצד במילון shared_courses שמכיל את הקורסים המשותפים ל-2 ההתמחויות.
3. סופרים את כל הקורסים שלא נמצאים ב2 ההתמחויות ומחשיבים אותם כבחירה מחוץ להתמחות.
4. עוברים על כל הקורסים שהם בחירה בהתמחות אחת אבל לא קיימים בהתמחות השניה:
   4.1. במידה ומספר נקודות הזכות שלקח הסטודנט בהתמחות המסוימת אינו עונה על הדרישה בהתמחות זו, נוסיף אותו לחישוב הנקודות של הקורסים באותה התמחות.
   4.2. במידה ומספר נקודות הזכות שלקח הסטודנט בהתמחות המסוימת עונה על הדרישה בהתמחות זו, נוסיף אותו לחישוב הנקודות של קורסי בחירה מחוץ להתמחות (ולא למילון shared_courses מאחר והקורס נמצא התמחות אחת אבל לא בשניה).
5. מוסיפים את כל הקורסים שהם בחירה ב2 ההתמחויות למילון shared_courses.

לאחר המיון האחרון שמתבצע נעבור על המילון shared_courses ונתבסס על וריאציה שלנו לבעיית תרמיל הגב על מנת להתאים את הקורסים שנותרו לכל סוג התמחות:

1. יוצרים שקים בגודל של כמות הנק"ז שחסרה בהתמחות (ובבחירה מחוץ להתמחות).
2. מיין את השקים בסדר עולה לפי הנפח
3. כל עוד לא סיימנו למיין את הקורסים הנותרים נבצע את הפעולות הבאות:
   3.1. בחר קורס.
   3.2. הגדר את הbenefit של כל שק כנפח הנוכחי שלו חלקי הנק"ז של הקורס
   3.3. צור רשימה ממוינת בסדר יורד של הbenefits
   3.4. עבור כל שק מהרשימה הממוינת, אם הנפח שלו קטן מהנק"ז של הקורס:
      3.4.1. הכנס לשק את הקורס והוצא אותו מרשימת הקורסים הנותרים.
      3.4.2. עדכן את הנפח שלו.
   3.5. אם הקורס לא נכנס לאף שק, העבר אותו למילון נפרד ששומר קורסים שלא נכנסו והסר מרשימת הקורסים הנותרים.

4. העבר את הקורסים מהשקים להתמחות המתאימה (או בחירה מחוץ להתמחות)
5. עדכן נק"ז חסר.
6. אם ישנם קורסים שלא נכנסו לאף שק (הם היו גדולים יותר מאפשר היה להכניס), הקצה אותם אחד אחרי השני למקומות שחסר. אם לא חסר באף מקום, הקצה אותם להתמחויות עד שיגמרו הקורסים.

לאחר בדיקת קורסי ההתמחויות, בודקים האם נלקחה כמות מספיקה של קורסי חובה ושל נקודות זכות (חישוב סה"כ הנקדות שנלקחו ובדיק אל מול כמות הנקודות הדרושה לסיום התואר).