ב"ה

תיעוד לתרגיל בית הדסים 5.0

מגישה: אורה מויאל

חלק א

תרגיל א:

הקוד לתרגיל ממומש בפייתון, בקובץ exercise_A1.py.

מימשתי את הפונקציה n_most_frequent_errors. הפונקציה קוראת את הנתונים מהקובץ בצורה מפוצלת באמצעות ספריית itertools, כל פעם קראתי 1000 שורות מהקובץ.
הפונקציה סופרת את שכיחות השגיאות בכל חלק, ואז ממזגת את השכיחויות של השגיאות.
בסוף היא מחזירה את n קודי השגיאה הנפוצים ביותר.

סיבוכיות זמן ריצה עבור קלט באורך n, E סוגים של קודי שגיאה וN קודי שגיאה מבוקשים:

- Θ(n) מעבר על כל הדיווחים.
- O((n/1000)*E) מיזוג של המילונים שמונים את השגיאות.
- Θ(NlogE) מציאה של N קודי שגיאה שכיחים.
- Θ(N) החזרה של קודי השגיאה השכיחים עם השכיחות שלהם.

סה"כ: Θ(n/1000)*E)

סיבוכיות מקום בזיכרון:

- Θ(n) טבלה של הנתונים
- O((n/1000)*E) רשימה של מילוני שכיחות של השגיאות
- O(E) מילון של שכיחויות

סה"כ: O((n/1000)*E)

ב"ה

תרגיל ב

הקוד לתרגיל ממומש בפייתון, בקובץ exercise_A2.py.

מימשתי את הפונקציה mean_fer_hour שקוראת את הנתונים לדאטהפריים בספריית pandas ומחשבת ממוצעים לפי שעות. בהתחלה הפונקציה בודקת את תקינות התאריכים, ומוחקת דיווחים כפולים, דיווחים עם תאריך שלא קיים בלוח השנה, ודיווחים ללא ממוצע. לאחר מכן הדאטה מחולק לפי ימים, מחשבים ערך ממוצע לשעות שבכל יום, ואז ממזגים את השעות של כל היום. את הפלט מייצאים לקובץ CSV.

כשהנתונים מגיעים בזרימה אפשר לשמור כל פעם את מספר ערכים שקיבלנו עד עכשיו בשעה הנוכחית, וגם את סכום הערכים. כל פעם שמקבלים ערך מוסיפים אותו לסכום הערכים, ומעלים ב1 את מספר הערכים. אחרי כל עדכון אפשר להציג ממוצע שעתי- מחלקים את סכום הערכים מהשעה הנוכחית למספר הערכים שהתקבלו בשעה הנוכחית. כל פעם שמתחילים שעה חדשה מאפסים את הסכום והמונה של הערכים שהתקבלו.

התאמתי את הקוד לקריאת נתונים מקובץ פרקט (parquet).

פרקט הוא קובץ שמאורגן לפי עמודות, ולא לפי שורות כמו קבצי CSV. היתרון שלו זה שאפשר לטעון רק עמודות מסויימות מהקובץ אם צריכים רק חלק מהמידע על כל הרשומות.

בקובץ פרקט סוג הנתונים שבעמודות מוגדר מראש, ולא צריך לקבוע אותו כשקוראים את תוכן הקובץ.

בנוסף הדחיסה של הנתונים בקובץ יעילה בהשוואה לקבצים אחרים בגלל ששומרים ביחד נתונים מאותו סוג ביחד

ב"ה

חלק ב

הקוד לתרגיל ממומש בפייתון, בקובץ exerciseB.py.

מימשתי את הפונקציה connections שקוראת את הקובץ עם המידע האישי ויוצרת טבלה עם הקרבה בין האנשים. בנוסף הפונקציה משלימה בני זוג גם כאשר הנתונים חלקיים.

כדי לבדוק את הקוד יצרתי טבלת דמה עם נתונים על אנשים בקובץ personal_data.csv.

חלק ג

1) רמת העניין שלי בתפקידי חומרה היא 1, לא מעוניינת.
2) השלט משדר הבהובים של קרינה אינפרה אדומה- רצף שמסמלים מספרים בינאריים, כאשר אור מסמל 1 והיעדר אור מסמל 0.
בצד השלט צריך להיות רכיב שמקודד את פקודות להבהובים ורכיב שממשדר את הקרינה. בצד של המזגן צריך להיות רכיב שקולט את הקרינה ורכיב שמעבד את האותות לביצוע של הפעולות הדרושות.
המזגן מזהה את הפקודות שהוא צריך לבצע לפי הקידוד הבינארי של הפקודות.

ב"ה

חלק ד

בניתי מערכת להזמנת סחורות לחנות. הנתונים נמצאים בmongoDB והbackend ממומש בפייתון בקובץ exerciseD.py.

פעולות אפשריות במערכת באמצעות API:

ספקים:

- register_supplier- הרשמה במערכת באמצעות שם חברה, שם נציג, מספר טלפון ורשימת מוצרים.
- login_supplier- התחברות באמצעות שם חברה ושם נציג.
- view_orders_for_supplier- צפייה בהזמנות שהתקבלו לספק.
- approve_order- אישור של הזמנות- שינוי של הסטטוס של ההזמנה ל"בתהליך".

חנות:

- order- הזמנה של רשימת מוצרים מספק.
- view_orders- צפייה בכל ההזמנות שלו הושלמו.
- complete_order- השלמת הזמנות- שינוי של הסטטוס של ההזמנה ל"הושלמה".