<u>תכנות מונחה עצמים מטלה 0 (חלק 1):</u>

חלק 1: מידול בעיה, תכנון, דיאגרמת מחלקות.

חלק 2: מחלקות, ממשקים אבסטרקציה

מטלה זו מוקדשת לחזרה על נושאים שנלמדו בקורסים קודמים של מבוא לחישוב, ומבנה נתונים.
בניגוד למטלות הבאות מטלה זו היא בעברית, ומוגשת דרך המודל (כל המטלות הבאות תהנה באנגלית והן יוגשו כפרויקט ב Github).
במטלה זו נפתח אלגוריתמיקה עבור מערכת מעליות של בניין רב קומות. כללית נחשוב על בניין (Building) כבעל קומת מינימום (יכולה להיות שלילית) וקומת מקסימום, כאשר כל קומות הביניים קיימות. בכל בניין יהיו מספר מעליות (אחת או יותר). לצורך הפשטות נניח שכל מעלית (Elevator) יכולה להגיעה לכל אחת מהקומות. מעבר לכל לכל מעלית יש מאפיינים של זמן עצירה, זמן תחילת תנועה, ומהירות (כמה קומות עוברת המעלית בשנייה).

כללית נחשוב על המעליות בבניין כ״מעליות חכמות״ משמע: המשתמש צריך פשוט להקיש מחוץ למעליות את קומת היעד (destination) ואז המערכת צריכה לשבץ (להקצות) לו מעלית מסויימת אשר מוגדר לה כבר לעצור בקומת היעד.
במטלה זו האתגר נתמקד באתגר האופטימיזציה הבא :
בהינתן קריאה למעלית מקומת המקור לקומת היעד - המערכת תרצה לשבץ את המעלית שתצמצם למינימום את זמן ההגעה (זמן ההגעה מוגדר להיות משך הזמן בשניות שבין הקריאה למעלית ובין ההגעה לקומת היעד). באופן יותר כללי נאמר שבהינתן אוסף של קריאות למעליות בזמן נרצה להגדיר אסטרטגיית שיבוץ מעליות לקריאות שתצמצם למינימום את סך משך זמן ההגעה עבור כלל הקריאות.

הנחיות והדרכה:

- חשוב להכיר את מרחב הבעיה - הסתכלו סביבכם ונסו להבין כיצד מעליות חכמות עובדות. זכרו: מדובר במעליות ״חכמות״ בהן בחירת קומת היעד מבוצעת בקומת המקור - מחוץ למעליות (אין אפשרות לבחור את קומת היעד בתוך המעלית).
- לפני שאתם מתחילים לתכנן (והרבה לפני שמתכנתים), חשוב מאוד לשבת ולהבין את מכלול מרכיבי הבעיה שרוצים לפתור - במערכות מורכבות זו למעשה המשימה העיקרית - לאחר שתבינו היטב את הבעיה ותמדלו אותה באופן מדויק, תוכלו לייעל מאוד את תהליך הפיתוח והמימוש של המטלה (הערה זו נכונה לכל המטלות בקורס). מומלץ להסתכל על הסרטונים הבאים, שסוקרים את הבסיס של האלג׳ של מעלית (על מעליות ישנות): https://www.youtube.com/watch?v=BCN9mQOT3RQ
https://www.youtube.com/watch?v=oY1QlCqWOss
- למען הפשטות נגדיר שבכל הבניינים שלנו, הקומות שערכן קטן או שווה לאפס מהוות קומות כניסה או יציאה מהבניין, וכל הקומות שערכן 1 ומעלה מהוות קומות (מגורים או משרדים) ללא יציאה מהבניין. נניח שבכל בניין קומת המינימום קטנה שווה ל 0 וקומת המקסימום גדולה שווה ל 1.
- באופן טבעי ניתן לחשוב שרוב הזימונים של מעליות הן מאחת מקומות הכניסה לאחת מקומות המגורים (או ההיפך). אבל יתכנו גם זימונים בין קומות כניסה או בין קומות מגורים.
- ניתן לחשוב שכל מעלית יכולה (אם כי לא חייבת) לעצור בכל קומה. ואין הגבלת כמות אנשים לכל מעלית.
- כל מעלית יכולה להיות במצב ״עלייה״, ״ירידה״ או ״מנוחה״. אם לא הוקצתה למעלית שום קריאה היא נשארת במקום במצב ״מנוחה״. כאשר המעלית מקבלת זימון (משובצת לה ״קריאה״) היא צריכה להגיעה לקומת הקריאה (המקור) של הזימון ולאחר מכן להגיע לקומת היעד. נגדיר זימון להיות ״עולה״ אם קומת המקור נמוכה מהיעד (והפוך לגבי זימון ״יורד״).
- בפתרון המלא (אלג׳ online) ברגע שמתקבלת קריאה (זימון) יש לשבץ עבורו באופן מידי מעלית, ולא ניתן לשנות אותו לאחר מכן, ובהתאם לשלוח את המעלית לקומת המקור והיעד.

- ניתן לעצור את המעלית בקומת ביניים, לדוגמא אם המעלית עולה מקומה 1 לקומה 12 ניתן לעצור אותה בקומה 5 אם היא נמצאת בקומה קטנה או שווה ל 5, אבל לא אחרי זה.

מתחילים לעבוד "תכלס":

מטלה זו מתחלקת לשני חלקים (שבוע כל אחד): חלק ראשון היכרות ותכנון (ללא מימוש), חלק שני: מימוש המערכת. מטרת הפרויקט היא לחייב אתכם לחשוב קודם על מידול הבעיה, על האופן הנכון ולייצגת את מרכיבי הבעיה, וכמובן לתכנן את האלגוריתמים הנדרשים.

שלבי עבודה - חלק א' ללא קוד:

1. סקר ספרות: התחילו בלהבין היטב את מרחב הבעיה, חפשו עבודות דומות שעוסקות באופטימיזציה של מעליות - ציינו לפחות שלוש עבודות שנראות לכם הכי רלוונטיות למטלה זו.
2. עמדו על ההבדל בין אלגוריתם online, לאלגוריתם offline. ההבדל נעוץ בזמינות הקלט, בעוד שבאלג' רגיל (off-line) כל הקלט זמין לכם מראש, ואתם יכולים להשתמש בכולו לצורך ביצוע החישובים, באלג' on-line הקלט מגיע כזרם נתונים בזמן, ועלינו לחשב תשובות על בסיס המידע שקיים לנו בכל זמן נתון. לאחר מכן נסחו אלגוריתם off-line עבור בעיית השיבוץ של קריאות למעליות - השתדלו לנסח את האלגוריתם באופן המדויק והפשוט ביותר להבנה.
3. נסחו אלגוריתם on-line בור בעיית השיבוץ של קריאות למעליות - השתדלו לנסח את האלגוריתם באופן המדויק והפשוט ביותר להבנה.
4. הגדירו את המחלקות העיקריות (ללא מימוש - רק דיאגרמת מחלקות) הנדרשות לכם לפתרון, לצורך כך ניתן להסתכל על החלק השני של המטלה.
5. חשבו על בדיקות פשוטות JUnit שניתן לעשות למערכת שתכננתם (ללא צורך במימוש) - הסבירו כיצד ניתן לבדוק את המחלקות שתכננתם.

סכמו את כל חמשת השלבים למסמך PDF (באנגלית או בעברית), רישמו את מספרי ת"ז של המגישים בראש המסמך והגישו למודל.

נספח א' - מבנה הקלט והפלט.

הכנו עבורכם מספר קבצי דוגמא שמייצגים קריאות למעלית בבניין, נסתכל על הקובץ הראשון (הקצר ביותר בין 10 שורות) שהיא בפורמט: csv (ניתן לפתיחה ב Excel),
לפי עמודות: שם, זמן (Time), מקור (s), יעד (d), דגל-סטטוס (f), ושיוך מעלית (תוצאת החישוב הנדרשת).
למעשה רק שלוש העמודות של זמן, מקור ויעד הן רלוונטיות כקלט - הסיבה שיש גם שתי עמודות נוספות (דגל סטטוס f, ותשובה נדרשת נועדה לאפשר קלט ופלט פשוטים ובעלי פורמט זהה - דבר שיקל בבדיקות.
במילים אחרות הפלט המרכזי של התוכנית שלכם הוא קובץ באותו פורמט של הקלט רק שהעמודה האחרונה (הימנית ביותר - שכעת היא כולה -1) תשתנה להיות התשובה שלכם לאיזה מעלית לשבץ כל "קריאה".

Name(ignore), Time (in seconds), s,d, f, a

```
Elevator call,0.437472729039321,0,1,0,-1
Elevator call,2.2094056301593525,0,1,0,-1
Elevator call,18.77299317966884,1,0,0,-1
Elevator call,19.590869649656987,1,0,0,-1
Elevator call,29.579433566814906,1,0,0,-1
Elevator call,33.242673193933555,1,0,0,-1
Elevator call,52.96277472128405,1,0,0,-1
```