תכנות מונחה עצמים מטלה 0 (חלק 1)-מעליות חכמות עם בקרת יעד .

מגישים :
יעקב חודורקובסקי - 207045063
ניר מאיר -313229106 .

1.מאמרים :

1. https://peters-research.com/index.php/papers/etd-algorithm-with-destination-dispatch-and-booster-options/
   ETD Algorithm with Destination Dispatch and Booster Options
   - שקלול זמן הגעה עבור מעליות חכמות

2. https://github.com/do-ryan/destination-dispatch-elevator-simulation/blob/master/A%20Heuristic%20for%20the%20Situational%20Application%20of%20a%20Destination%20Dispatch%20Elevator%20System.pdf
   A Heuristic for the Situational Application of a Destination Dispatch Elevator System
   - עוסק בהשוואה בין מעליות "פשוטות " לבין מעליות חכמות .
3. https://global.ctbuh.org/resources/papers/download/1050-elevator-selection-with-destination-control-system.pdf
   Elevator selection with destination control system
   האלגורתים של חברת המעליות KONE הגרמנית

סעיפים 2-3 . הקדמה , האלגוריתם לחישוב המסלולים , מצב Offline ומצב Online

הקדמה

מעליות עם בקרת יעד נועדו בשביל אופטימיזציה של שימוש במעליות בבניינים רבי קומות , המקבצת נוסעים לאותם יעדים לאותן מעליות, ובכך מצמצמת את זמני ההמתנה והנסיעה בהשוואה למערכת מסורתית שבה כל הנוסעים המעוניינים לעלות או לרדת נכנסים לאותה מעלית ו ואז לבקש את יעדם.
בניגוד למעליות הרגילות בה ידוע אך ורק כיוונו של הנוסע ולא איזה קומה הוא מגיע מראש , במעליות עם בקרת יעד , באותו רגע שבו המשתמש ניגש להזמנת המעלית אנו יכולים לחשב עבורו את המסלול האופטימלי מכיוון שאנו יודעים את המוצא היעד ישירות .

חישוב מסלול

במקביל למעלית רגילה (לא חכמה) מכיוון שידוע מראש נקודת ההתחלה והסיום של הנוסע, האלגוריתם יכול לבחור עבור הנוסע את המסע המהיר ביותר.
המסע מתחיל מאותו הרגע שבו הוא הזמין את המעלית , כולל זמן ההמתנה אליה והנסיעה בה עד להגעה ליעד.

זמן המסלול מחושב על ידי :
המרחק בין הקומה נוכחית של המעלית (LEVEL) לבין קומת ה SRC + כמות העצירות.

פסודו קוד עבור תכנון עלות זמן הנסיעה עבור מעלית כלשהי :

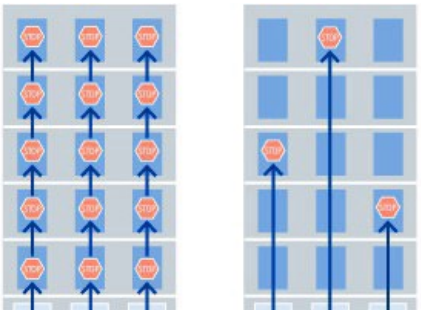
```
1.CostOfTravel (Elevator, startpoint, destination)
2.WaitingForList  = NumOfLevelstoTravel( current ,pickup );
3.RidingOnLift = NumOfLevelstoTravel( pickup, destination );
4. Stops  =countStopsinThatElivator * StopCosts ;
5.return WaitingForList + RidingOnLift + Stops ;

-- NumOfLevelstoTravel = abs ( from - to ) ;
```

מעלית תמיד תמשיך לעלות/ לרדת לכיוון הקומה המיועדת מבלי לשנות כיוון בדרך (אך יכולה לאסוף אנשים נוספים לאותו כיוון) עד שיגמרו הבקשות לכיוון
ואם אין בקשות נוספות , אז המעלית תיכנס למצב מנוחה או תלך לכיוון קריאה אחרת .

עבור אלגוריתם Offline

מכיוון שיהיה ידוע לנו מראש את כמות הנוסעים נוכל לדעת את מסלולם מראש , נוכל לייעל את זמן הנסיעה בכך ש מראש נוכל לבחור מעלית שתצא לאותו האזור .
לדוגמא , אם בבוקר רוב האנשים יוצאים מקומת הקרקע משרדיים , אז המעליות יחכו למטה , ויחלקו בינהם את הקומות לפי כמות המעליות ויורידו את הנוסעים באזורים (ראו איור 1) . ואותו דבר עבור סוף יום , מכיוון שבסוף היום רוב הביקוש מגיע מהקומות של המשרדים אל קומת הקרקע , שם יחכו להם המעליות מראש.



בנוסף , כמות הניסעות הכללית נוכל לשאוף לחילוק שיוויוני בין המעליות .

<u>איור- 1</u>

<u>ניתן לראות שאנו יכולים לנצל את היתרון שאנו יודעים את היעד</u>

עבור אלגוריתם Online

שהאלגוריתם רץ במצב Online אז אנו מקבלים קריאות בזמן אמת , לכן בכל רגע נתון , המעליות שולחות מידע לבקרה היכן הם נמצאות , ולאן פניהם מועדות .
בזמן שמגיע עובר אורח ומזמן מעלית לקומת היעד אליה הוא רוצה להגיע , הקריאה נכנסת לבקרה והמערכת מחשבת את עלות הזמן עבור כל מעלית בנפרד , ומזמינה עבורו את המעלית האופטימלית ביותר.

עיכובים שיכולים להיווצר בזמן אמת :

- נוסע שנכנס למעלית עבורו תוכנן זמן מסוים , זמנו יכול להתארך מכיוון שעוד נוסע הצטרף אליו בדרך.
- עיכוב נסיעה יכול להגיע תוך כדי המתנה, לדוגמא : בזמן שמישהו מחכה למעלית שיועדה עבורו בזמן אופטימלי, אז מישהו יכול להצטרף למעלית זו בדרכה למטה, ובכך להאריך את הזמן של ההגעה עובר הנוסע שמחכה.
- בזמן שכל המעליות נמצאות במצב עליה, <u>וכולן</u> נמצאות מעל קומת ההזמנה, אז הנוסע שלמטה ימתין עד שאחת מהן תסיים את המסע שלה .
- אם לדוגמא מישהו מזמין מעלית מ 1 לקומה 7 , ואז אחריו יבוא מיד נוסע אחר וירצה לעלות לקומה 8 לדוגמא , אז המערכת תעדף עבורו אם קיימת מעלית ריקה שנמצאת איתו באותה הקומה מאשר המעלית המיועדת לקומה 7 . מה שיגרום ל 2 מעלות לעלות למעלה , ולהזמנות עתידיות מה שיכול לגרום לעיכובים .

ההבדל בין אלגוריתם OffLine ל OnLine

אלגוריתם Offline ידוע לנו מראש את כל המסלולים הקיימים במעליות עבור הנוסעים , ולכן זמן הנסיעה ידוע מראש עבור כל נוסע ונוסע שהולך להיכנס למעלית, ובכך נוכל גם להתאים את המעליות שיחכו מראש לנוסעים אם אינם עובדות .

אלגוריתם Online בנוסף להזמנות הקיימות , מתקיימות הזמנות בזמן אמת , מה שיכול בכל רגע לשנות את עלות המסע של מעלית כלשהי , על ידי כך שנוסע אחר בחר במעלית זאת שהייתה יעילה עבורו , ולכן יהיה עלינו לחשב בכל פעם שמישהו מזמין מעלית את עלות הנסיעה הזמינה עבורו בכל מעלית אחרת .

דוגמא המציגה איך האלגוריתם בוחר מעלית עבור נוסע :

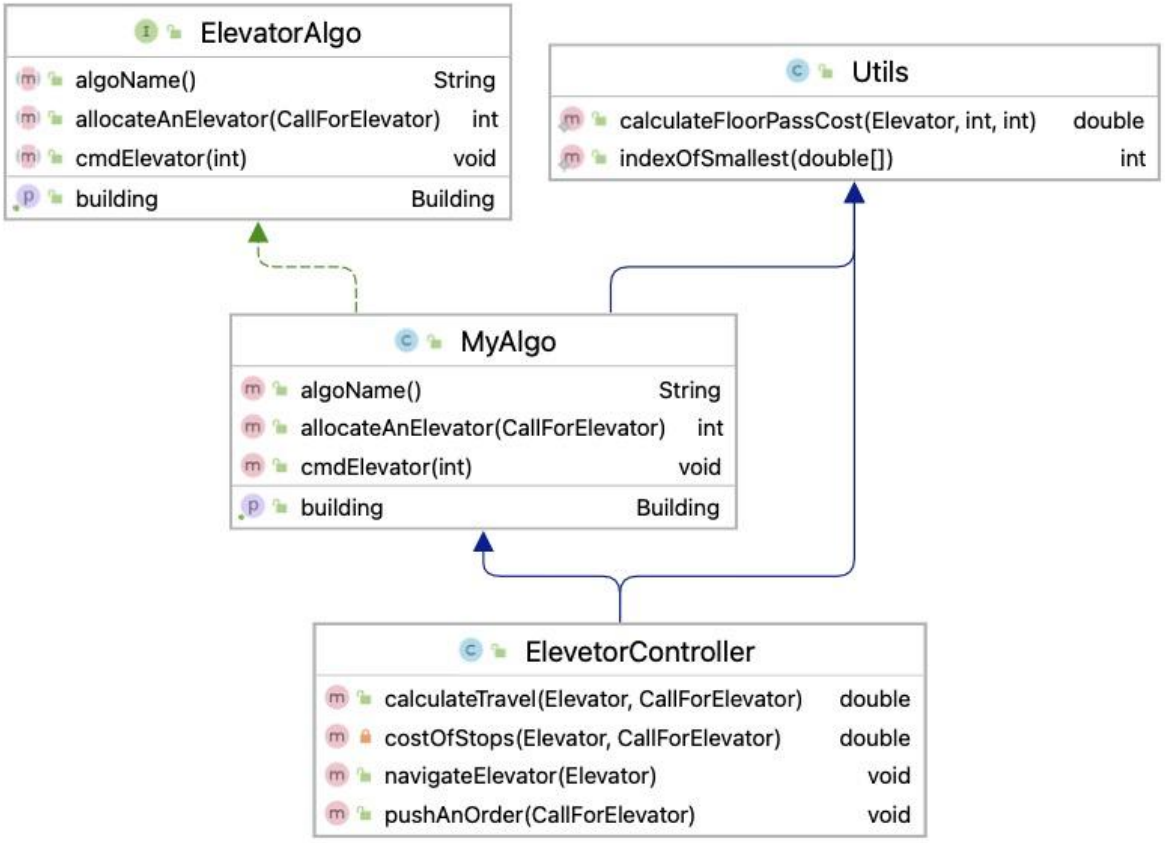
דניאל מגיע לבניין ורוצה לעלות מקומה 0 לקומה 8 , זמן עצירה הינו 10 שניות , זמן מעבר בין קומה לקומה הינו 3 שניות .

נניח כי בבנין יש 3 מעליות והן נמצאות במצבים מסוימים :

מעלית	1	2	3
נתונים	- קומה וכיוון : 7 , עולה - נוסעים : 1 עוצר בקומה 8	- קומה וכיוון : 6 , יורד - נוסעים 2 עוצר בקומה 4 עוצר בקומה 3	- קומה וכיוון : 5 יורד - נוסעים : 3 עוצר בקומה 4 עוצר בקומה 3 עוצר בקומה 2
עלות זמן נסיעה -שניות	דיליי הנוסעים - 10 להגיע לקומת הקרקע -27 עצירה ועלייה - 34 סה״כ 71 שניות	דילי הנוסעים - 20 הגעה לקומת הקרקע - 18 עצירה ועליה - 34 סה״כ 72 שניות	דילי הנוסעים - 30 הגעה לקומת הקרקע - 15 עצירה ועלייה - 34 שניות סה״כ 79

מכאן הבקרה תבחר את מעלית 1 עבור דניאל, אפילו שהמעלית קודם תעלה למעלה ואז תיקח אותו.

4. דיאגרמת מחלקות - UML



5. טסטים ל Junit

עבור הטסטים , נוכל קודם לעשות @before אשר מתגדיר מצב קיים בכמה מעליות, נוכל גם לתאר מצב offline יהווה תוכנית או שאפילו לראות איך המעלית מגיבה למצבים מסויימים באונליין .
את הטסטים נריץ על המבקרה, בכך שנכניס לה ערכים מאיזה ולאיזה קומות אנו רוצים להגיע , לראות שהמערכת בוחרת עבורנו את המעלית האופטימאלית .

בנוסף נוכל לבדוק שהמערכת יודעת לחשב נכון את עלות הנסיעה (עצירה , הגעה לקומת התחלת המסע והיעד) .

עוד מקורות אחרים

https://stackoverflow.com/questions/493276/modelling-an-elevator-using-object-oriented-analysis-and-design

שאלה מסטאק אובפלו .

https://towardsdatascience.com/elevator-optimization-in-python-73cab894ad30

מאמר צורך וכו ..

https://www.angelfire.com/trek/software/elevator.html

https://en.wikipedia.org/wiki/Elevator_algorithm ויקי

https://leetcode.com/discuss/interview-question/object-oriented-design/124936/design-an-elevator-system

https://leetcode.com/discuss/interview-question/559454/OOD-Design-elevator-system

https://github.com/seahrh/liftoo

https://leetcode.com/discuss/interview-question/object-oriented-design/846057/oodelevator-system-with-n-elevators

https://www.youtube.com/watch?v=siqiJAJWUVg

Lecture 9: Object Oriented Analysis and Design Part 3: Object Oriented Design of an Elevator

Git https://github.com/komidawi/ElevatorControlSystem