

Task 0 - Elevator Algorithm

סקר ספרות:

- (1) <http://www.columbia.edu/~cs2035/courses/ieor4405.S13/p14.pdf>
- (2) <https://towardsdatascience.com/elevator-optimization-in-python-73cab894ad30>
- (3) <https://www.popularmechanics.com/technology/infrastructure/a20986/the-hidden-science-of-elevators/>

אלגוריתם offline

(1) כל המפלגות במבצע התחלת כרזע בקומת הכניסה.

(2) מפלגת מס' 1 - תאסוף את האנשים שבוצים לפלגות ותזעזע לקומה הכי גבוהה שרצו לפלגות אלה.

(2) * מפלגת מס' 2 - תוריד את כולם לקומה הכי נמוכה שרצו להגיע אליה.

(3) לאחר סיום של מס' 2 - מפלגת מס' 1 תזעזע לקומה הכי גבוהה שממנה רוצים לרדת ובדרך תאסוף את כל מי שחצה לרדת.

(3) * מפלגת מס' 2 - תזעזע לקומה הנמוכה ביותר שממנה רוצים לפלגות ובדרך תאסוף את כל העולים.

(4) מפלגת מס' 1 - תעשה את מה שמפלגת מס' 2 עשה. במידה ויש מפלגת אחת היא תבצע גם את *.

(5) אם קיימות עוד מפלגות נשלח אותם לפי סדר הבא: מפלגות עם מס' אי-זוגי למטה, מפלגות עם מס' זוגי למטה

אז מנע להפחית את העומס.

תחילה נגדיר איחוד באופן הבא:

כאשר משהו מקיף קומה וצויה, אלונו למצוא לו מעלית מתאימה.

הכסיה שלנו זה להביא פתרון לקריאות שקומות בזמן אמת.

הקדמות: למעלית יש 3 מצבים: עלולה, יורדת, עומדת (לא בתנועה).

הערת קריאה הערכה הדיברים הבאים:

1) המערכת תקבל קריאה ותשמור את הקומה שממנה תבצע הקריאה ותשמור את הקומה שאליה חוצים להגיע.

2) א. המערכת תדבר על כל המעליות שקיימות ותבדוק אם קיימת מעלית שהיא כבר בדרכ לקומה שממנה

תבצעה הקריאה. כמו כן, היא גם תבדוק האם המעלית עלולה/יורדת והאם זה עומד להגיע לקריאה.

ב. אם קיימת מעלית כזאת היא תאסוף את המשתמש ותקח אותו לקומת היעד שלו.

3) א. אם שלא 2) לא הצליח להתאים מעלית לקריאה נחפש מבין המעליות שנמצאות במצב עמידה.

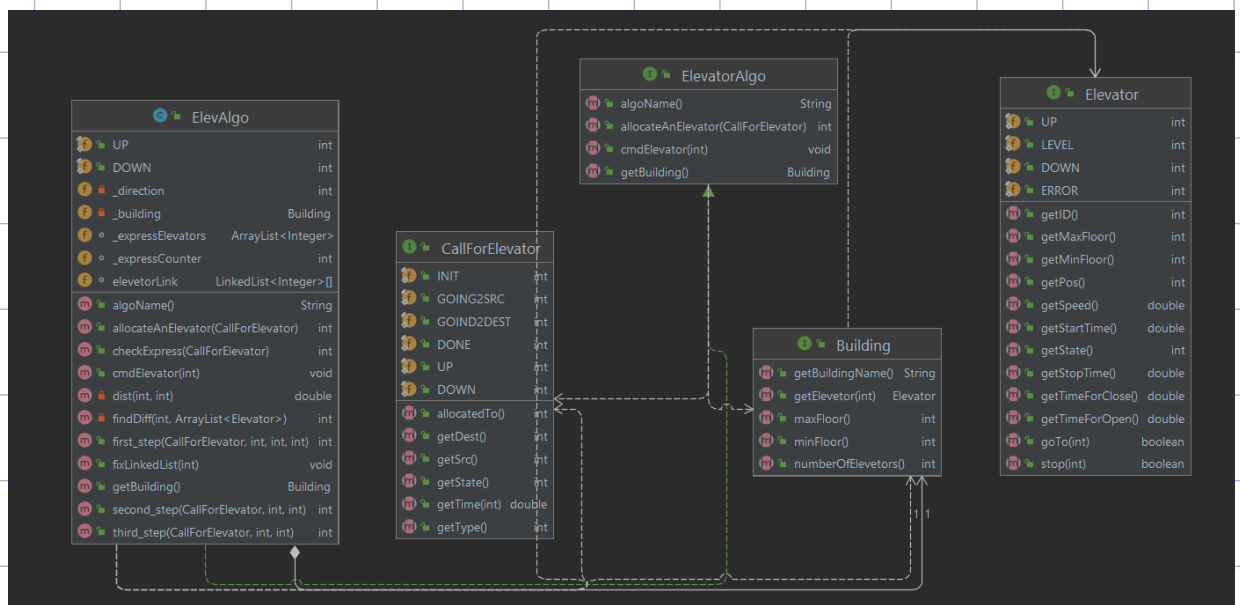
ב. מבין כל המעליות שנמצאו נבחר את המעלית שצריכה להגיע הכי מהר לקומת הקריאה.

4) א. במידה ושלבים 2) ו-3) לא הצליחו למצוא התאמה, נסתכל על כל המעליות שכן הם כולם בתנועה.

ב. נחשב את הזמן המשוער לסיום המשימה הקודמת של כל מעלית בנוסף למרחק שלה מקומת הקריאה.

ג. ניקח את המעלית שיחס הזמן שלה נמוך ומרחקה מקומת האיחוד הוא הקטן ביותר ונתאים אליה לקריאה.

זיכרון מתחלקות



בדיקות JUnit אפשריות

פונקציית AllocateAnElevator:

נעשה יותר קריאות בשגייה מאשר במות המעליות.

נבדוק שברגע שאנו מקצים למעלית קריאה אנו לא מוחקים את הקריאות הקודמות שלה.

האם אנו מוצאים את המעלית האופטימלית עבור הקריאה.

מה קורה כאשר מקצעים מספר קריאות מאותה קומה לאותו הסד באותו הרגע.

פונקציית distance:

האם הפונק' מתשבת את המרחק בצורה הנכונה.

האם הפונק' משתמשת בכל נתונים הקשורים לחישוב מרחק זמן.

פונקציית ElevatorCmd:

האם הפונק' מוודא שהמעלית נפיימת את כל הקריאות.

האם הפונק' לא משנה כיוונים כאשר יש לה קריאות קיימות.

האם הפונק' לא מליצה מעליות סתם.