

# מטלה 0 חלק 1 - OOP

1. ספרות:

<https://www.youtube.com/watch?v=BCN9mQOT3RQ> <https://www.youtube.com/watch?v=oY1QICqWOss>

2. off-line אלגוריתם :

תחילה נבדוק את הכיוון של הזמנת המעלית, נבחין אם ישנן מספר מעליות בעלות אותו כיוון נבצע בדיקת "מקור יעד" - מאיפה כל מעלית יוצאת ולאן כל מעלית צריכה להגיע.

בהתאם לכך נבחר את המעלית שהיעד שלה ממוקם אחרי היעד שאנו צריכים – כך שתוכל "להוריד" אותנו בדרך.  
סביר להניח שיהיו מספר מעליות שינועו לאותו כיוון שאנו צריכים לפיכך נבצע השוואה בין הזמנים שלהם, זאת אומרת נבדוק מי מבין המעליות תגיע אל קומת "המקור" שממנה הוזמנה המעלית בזמן הקצר ביותר- הדרכים לבדוק זאת הם:

1. מיקום המעליות ביחס לקומת המקור.
2. חישוב זמן פתיחת+ סגירת דלתות של כל מעלית.
3. בדיקת מספר העצירות המיועדות עבור כל מעלית.

בנוסף לבדיקות אלו תבוצע הגבלת קריאות עבור כל מעלית ע"י חלוקת מספר הקומות במספר המעליות, כך נקבל את מספר המקסימלי של עצירות (וקריאות) של כל מעלית.

(מעלית שמכסת ההזמנות שלה מלאה תוסר מהאופציות של בחירת המעלית שאנו צריכים).  
את כל הבדיקות שצינו לעיל נבצע ע"י ריצה על מערך (מערך של מעליות) והשוואה פרטנית עבור כל בדיקה.

לסיכום האלגוריתם - נעבור על כל המעליות ונוודא מי מהן מפיקה את זמן התפעול הקצר ביותר – את המעלית הראשונה שאיתה נתחיל להשוות נקבל באופן רנדומלי כדי לאפשר תנועה חיה של מעליות – ואפשרות יותר דינאמית לפעולה.

קריאות שאינן יכולות להיות מטופלות יישמרו במבנה נתונים באפר ויוטפלו כאשר המעליות לא יהיו עמוסות.

3. on-line אלגוריתם :

נשתמש בתור עדיפויות – הטיפול בבעיה יתבצע בהתאם לחישוב הזמנים של המעליות-  
חישוב הזמן המקסימלי למעבר על כל הקומות + זמן הריצה על כל מערך המעליות.



5. Junit:

1. נבצע בדיקה עבור 2 קריאות on-line המתקבלות במקביל.
2. נקרא לקומה שלא קיימת.
3. בדיקה עבור קריאה למעלית עולה- ע"י מציאת המעלית הקרובה ביותר שתגיע אל הקומה בזמן המהיר ביותר.
4. באותו אופן של 3 - קריאת למעלית יורדת.