1. [Elevator System Design | Object Oriented System Design Interview Question - YouTube](https://www.youtube.com/watch?v=siqiJAJWUVg)

[OOD - Design an Elevator | Ted Wang (tedweishiwang.github.io)](https://tedweishiwang.github.io/journal/object-oriented-design-elevator.html)

[Design an Elevator System (Design forum at Coderanch)](https://coderanch.com/t/707819/engineering/Design-Elevator-System)

1. באלגוריתם offline אנו נקבל מראש את הקלט ורק אז נפעל. לכן האלגוריתם הנ"ל צריך לפעול בצורה הבאה:

אנו נקצה מעלית קרובה לקלט בקומה הגבוהה ביותר שמבקש לרדת. לאחר שהמעלית עוצרת בקומה נמוכה ממנה התקבל הקלט, היא תרד ותעצור בכל קומה שבה קיים קלט ירידה. המעלית תמלא את משימת הירידה הקצרה ביותר ואז תמשיך למלא את המשימה הארוכה יותר (לרדת לקומה נמוכה יותר מזו הנוכחית). ברגע שלא יהיו עוד משימות ירידה למעלית היא תפסיק את פועלה ותיכנס למצב המתנה עד לקבלת קלט חדש.

באותו האופן אנו נקצה מעלית קרובה לקלט בקומה הנמוכה ביותר שמבקש לעלות. לאחר שהמעלית עוצרת בקומה ממנה התקבל הקלט היא תעלה ותעצור בכל קומה שבה קיים קלט עלייה. המעלית תמלא את משימת העלייה הקצרה ביותר ואז תמשיך למלא את המשימה הארוכה יותר (לעלות לקומה גבוהה יותר מזו הנוכחית). ברגע שלא יהיו עוד משימות עלייה למעלית היא תפסיק את פועלה ותיכנס למצב המתנה עד לקבלת קלט חדש.

1. עבור אלגוריתם online אנו נקבל קריאה ונקצה מעלית לפי חישוב הזמן האופטימלי.

שלב א: נבדוק אם יש לנו רק מעלית אחת בבניין. במידה וזה המצב הקריאה תלך למעלית הנ"ל

שלב ב: נבצע מעבר על כל המעליות. בכל מעלית אנו נדמה את רשימת הקריאות שלה לאחר הוספת הקריאה הנוכחית. לאחר מכן אנו נחשב את המעלית עם הזמן הטוב ביותר ונקצה אליה את הקריאה.

* על מנת לבצע את שלב ב' יהיה עלינו לדמות את מעבר המעלית על כל הקריאות שלה. אנו נבדוק אם היא קרובה יותר לקומת src התחתונה (מתוך קריאות עלייה) או אם היא קרובה יותר לקומת src העליונה (מתוך קריאות ירידה). לאחר שבדקנו זאת אנו נדמה את הזמן שייקח למעלית להגיע לאותה הקומה ולאחר מכן נדמה את הזמן שייקח למעלית לעבור את שאר הקריאות (אם היא התחילה מהקומת העלייה הנמוכה למשל אז המעלית תמשיך בעלייה ותבצע את כל הקריאות עלייה שיש ברשותה. לאחר שתסיים היא תמשיך לקריאות הירידה שלה מהקומה העליונה ביותר שהגיע ממנה קריאת ירידה).

שלב ג': המעלית עצמה קיבלה קלט ואנו צריכים להפעיל אותה. על המעלית לבדוק אם לא קיים לה קלט (ואז לצאת מהתוכנית), או שקיים לה קלט יחיד ויהיה עליה לפעול לפיו בלבד (מעבר לsrc ואז מעבר לdest ולאחר מכן מחיקה של הקריאה מהרשימה). במידה והיו לה יותר מקריאה אחת היא תפעל באותו האופן שבו חישבנו את הזמן. המעלית תגיע לקומה הקרובה אליה ביותר מבין שתי האפשרויות הקיצוניות, תמשיך לפי סוג הקריאה למעלה או למטה ולאחר הסיום היא תמחק אתכל הקריאות שלה מאחר וסיימה איתם.

|  |
| --- |
| **fixedElevator** |
| +count: LinkedList  +ListOfCalls: LinkedList |
| +fixedElevator() |

|  |
| --- |
| **Elevator\_MyAlgo** |
| +up: integer  +Down: integer  +Level: integer |
| -building: Building  -list[]: fixedElevator |
| +Elevator\_MyAlgo(Building b) |
| +getBuilding()  +algoName()  +allocateAnElevator(CallForElevator call)  +cmdElevator(int elev) |
| -bestTime(CallForElevator call)  -checkTotalTime(Elevator e, LinkedList<CallForElevator> l, CallForElevator call)  -checkDirection(LinkedList<CallForElevator> calls, int current)  -findLowFloorUP(LinkedList<CallForElevator> calls)  -findHighFloorDOWN(LinkedList<CallForElevator> calls)  -checkAmountOfFloors(LinkedList<CallForElevator> calls, int direction)  -checkNextFloor(LinkedList<CallForElevator> calls, int direction, int current)  -checkTime(Elevator e, int start, int end)  -dist(int i, int j)  -cmdElevator(int elev) |

1. מבחינת בדיקות יהיה צורך לבדוק מקרי קיצון ולוודא שהקוד יודע לתת מענה אלים. לדוגמא:

* מה יהיה הפלט במידה וקיימת רק מעלית אחת
* מה יהיה הפלט אם קיבלנו קריאות ירידה ועלייה מכל הקומות
* מה יהיה הפלט אם קיבלנו רק קריאות ירידה
* מה יהיה הפלט אם יהיו רק קריאות עלייה