# Lift Simulation Problem

Author: Prakash.

## Overview

A **lift** (or **elevator**) is a form of vertical transportation between building floors. It has to be programmed for it to work. The problem is in developing a solution that is efficient enough to minimize the movement of the lift between the floors while it is being used and saves time and power consumption. We will be building a model that would be efficient than the existing solution and can relace the existing algorithm.

## Context

Building a model that is efficient enough so that time is saved and the power consumption is at most minimum. The model would be built with multiple lifts, mirroring an actual system. It would have multiple lifts and multiple floors.

The major focus would be on the following functionalities:

1. Algorithm to determine when a lift should go to which floor
2. How the doors would interact.
3. What would happen if a lift is overloaded?
4. How the algorithm will be affected if one/more lift(s) are out of order/ inaccessible.
5. How to enhance accessibility so that the lift may be used by everyone.
6. What should lift do when it is ideal for some time.

## Existing Solution:

There are many existing solutions for this problem, which have been created using different stacks and are quite complex solutions that exist.

## Proposed Solution:

The tech stack used is HTML, CSS, and vanilla JS. Here HTML is being used mainly for building a skeleton of the model. CSS is used to give the skeleton its good looks and for design purposes. JS is the main programming language to build an algorithm that would specify the major functionality of the lift model. Frameworks like react and other similar frameworks can be used in place of vanilla js as it is purely for learning purposes it would be done later.

## Alternative Solution:

Since this project is for learning and self-development, the focus is HTML, CSS, and JS. Alternate solutions may incorporate frameworks in the future.

## Testing:

The model is to be tested so that it would meet all the basic functionality checks. The whole model would be divided and tested as single units. Once unit testing is done the integration testing would be performed by integrating the units and then the whole system would be tested.

The following test case would be taken into consideration:

- The model would be tested for multiple floors and multiple lifts inputs.
- It would also cover edge cases such as What action to be taken when a user gives multiple inputs.
- What to do when the emergency button is clicked.
- Would it be accessible to all including physically challenged people?
- Should it allow a kid under 5 years or not and many more test cases it would cover.
- What action to be taken when there is no power.
- When the doors are locked then what to be done.
- If the lift reaches maximum weight carrying capacity would it respond?
- When there is a fire in the building would it be accessible or not.