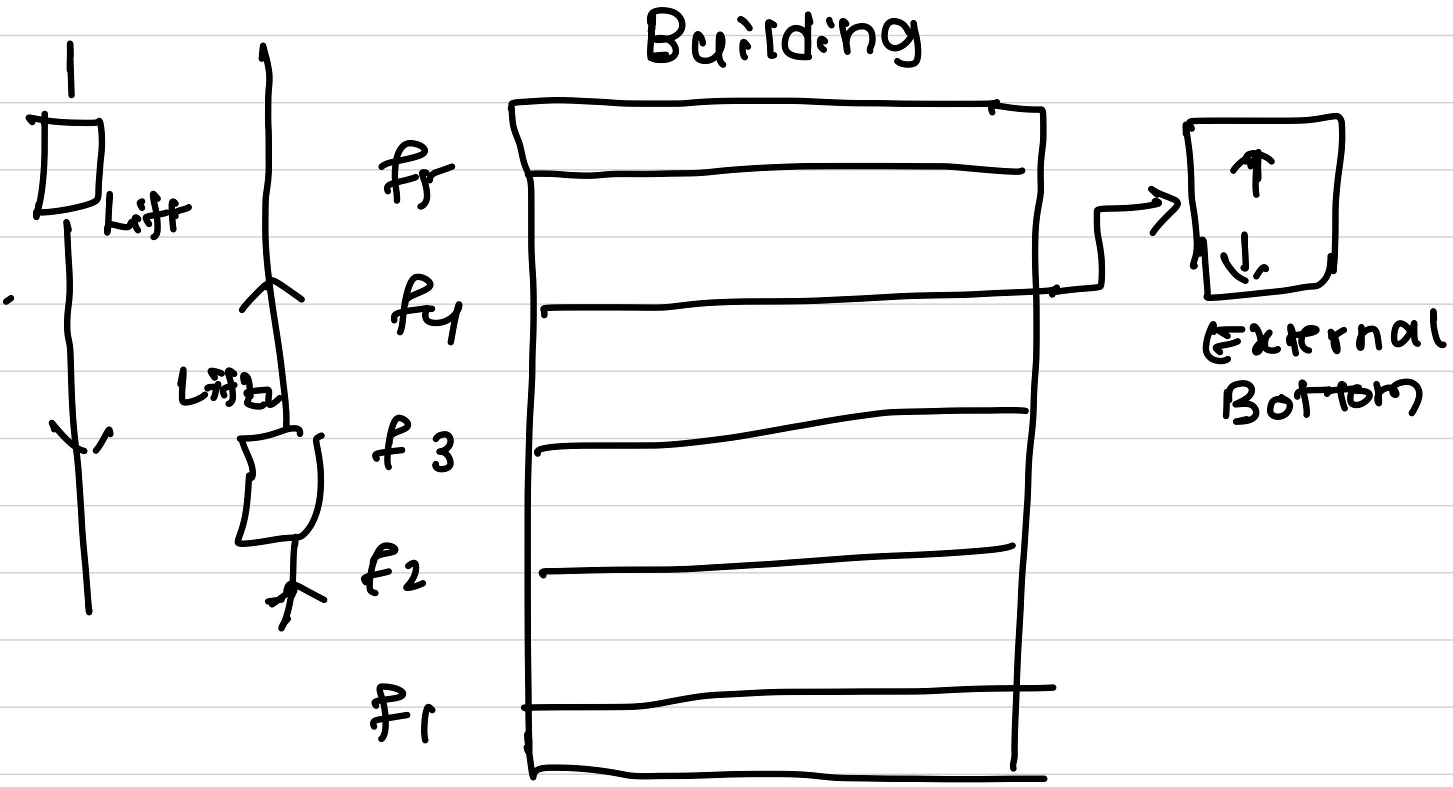
Elevator System

(Low Level Design)

Rough Flow:



1) User enters a Floor of a Building.

2) He clicks on External Button (either ↑ or ↓)

3) Elevator Car comes to the floor.

4) User enters the car and clicks on an internal Button

5) The car goes to the corresponding floor

Requirements:

1) How many no. of lifts ?? n

2) What is the algorithm for External botton? Can be

a) Even / odd

b) Fixed window for a particular lift

c) Minimum Seek Time

OBJECTS:

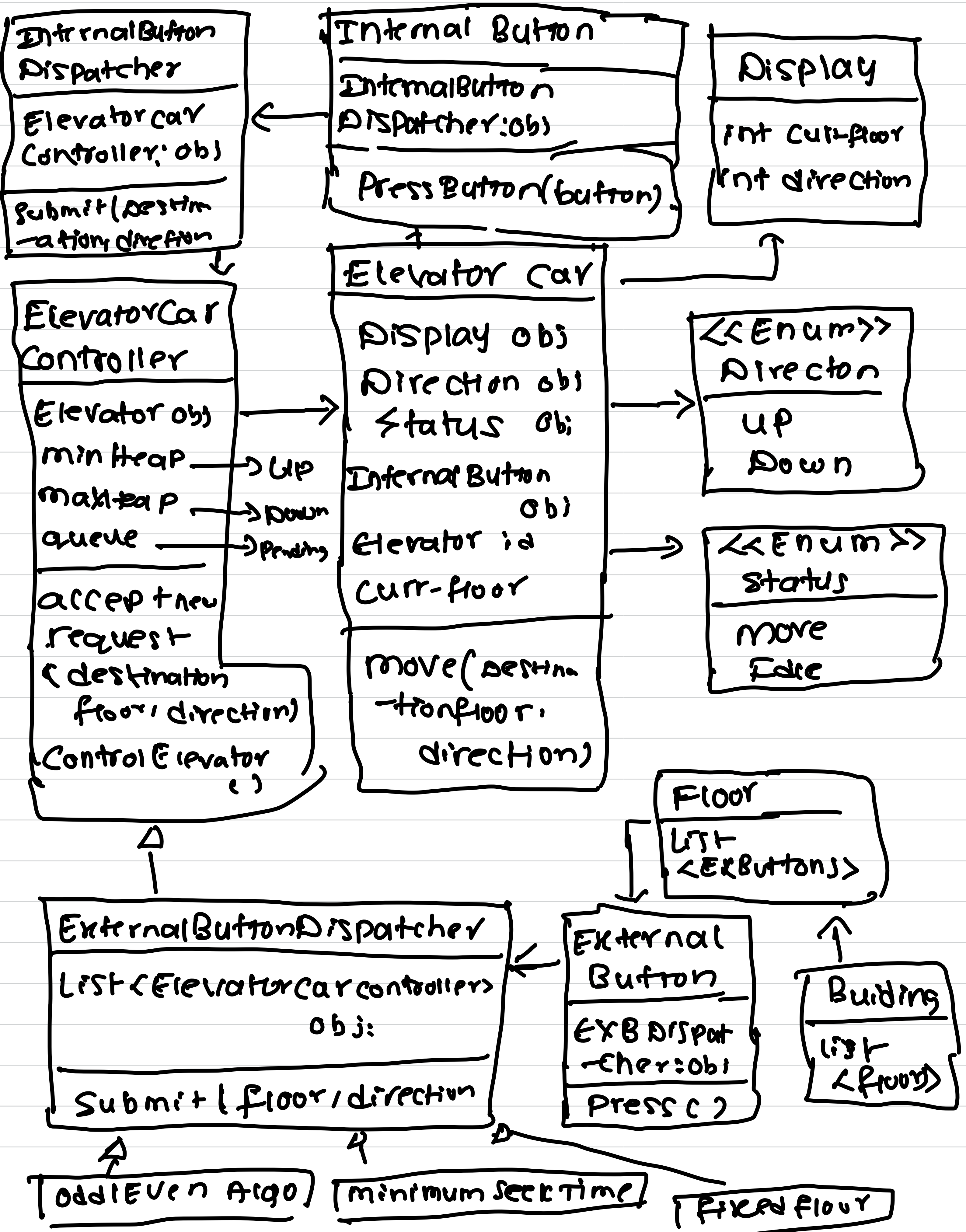
1) Building
2) Floor
3) External Button
4) Elevator car
   - Display
   - Direction (↑ or ↓)
   - Status (move or Idle)
   - Internal Button
   - Current floor
5) Display
6) Internal Button

Enums:

```
Direction {                Status{

    up,                        move,
    Down;                      Idle;
}                          }
```

Low Level Design: (Buttom Up Approach)



Algorithms:

1) External Button:

a) Even/Odd Algo

Some lifts support odd number floors and vice versa

b) Minimum Seek Time

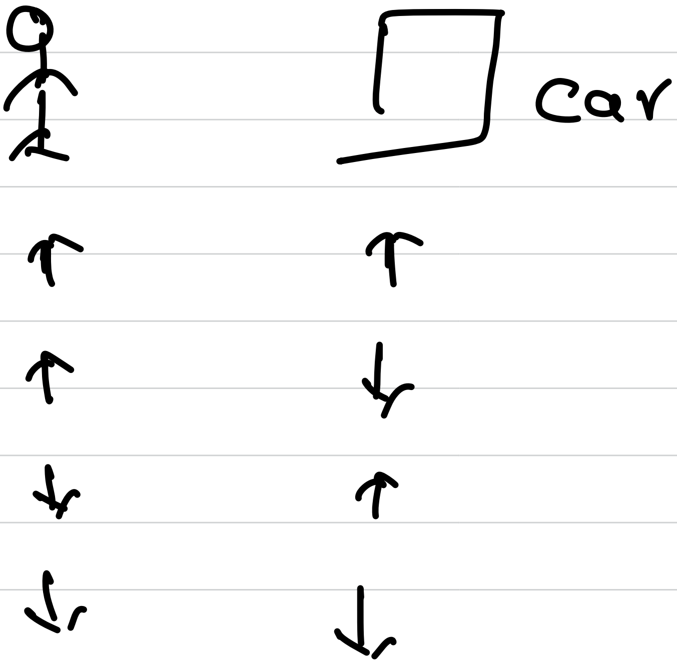
lesser time of arrival from the requested floor

c) Fixed window:

Some cars support only a set of floors

2) Elevator Algo: (Scan, Look)

Use cases :

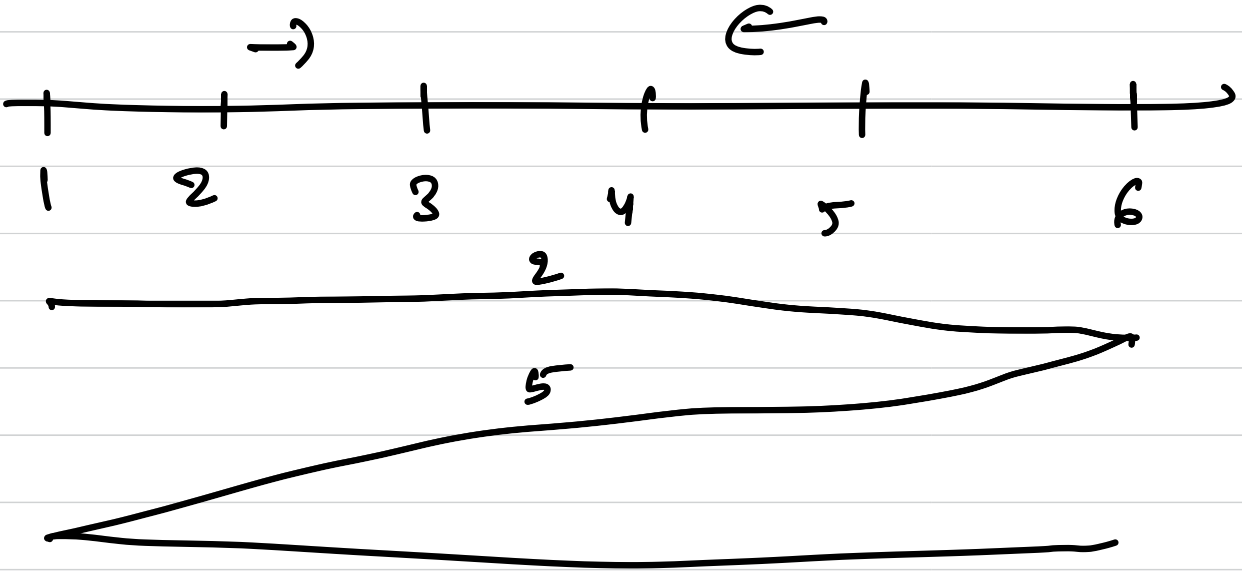


We use minHeap & maxHeap logic to support elevator algo.

Ex:

Current floor = 3

Scan

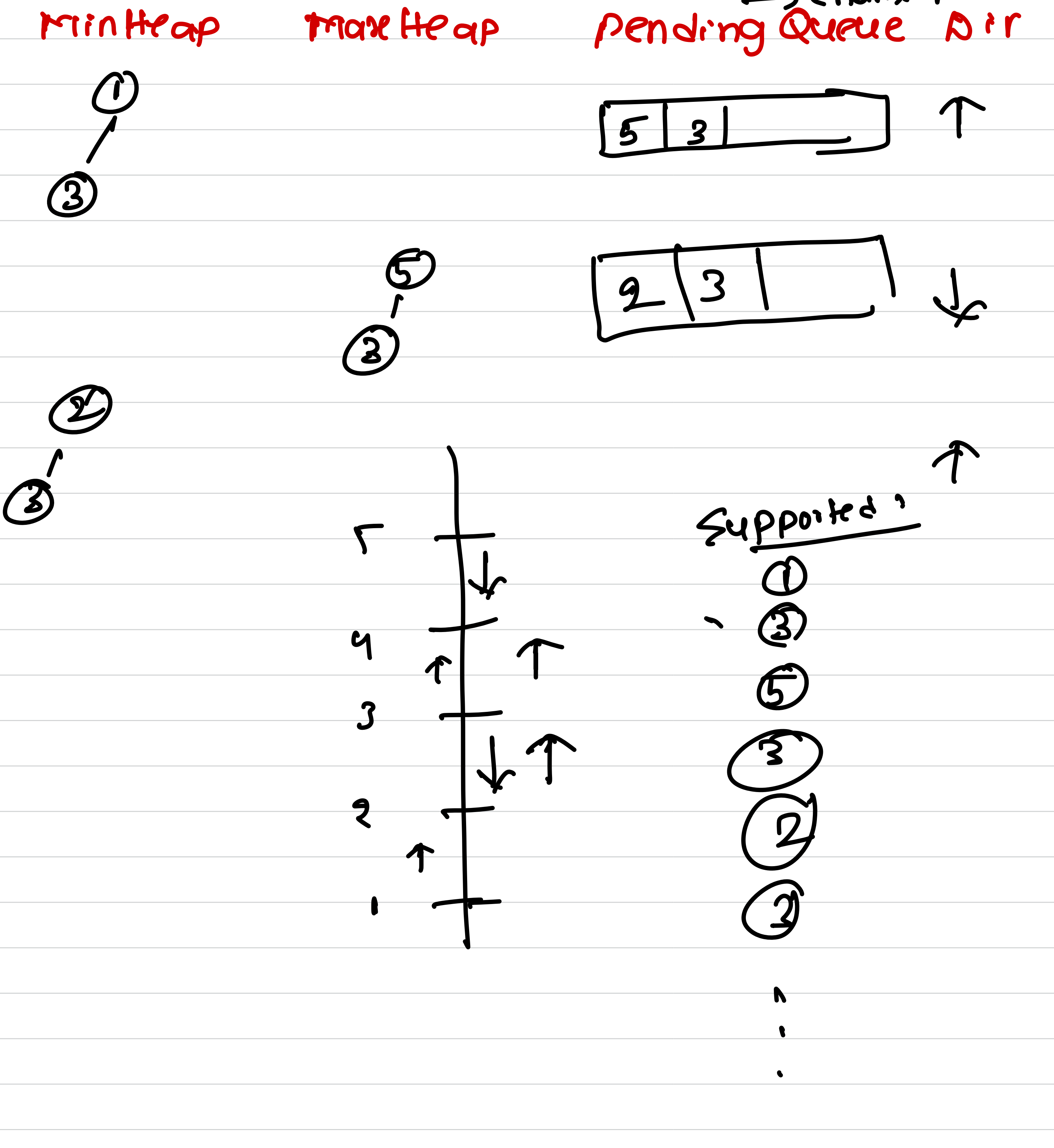


In Scan Algorithm, the car moves all the way from 0 to n-1 & vice-versa

Inefficient though ☹

∴ We came up with Look

Look: Move from current floor in a Direction
if ↑ then refer minHeap else maxHeap → make pa
→ change Directi



MinHeap
MaxHeap
Pending Queue
Dir
1
3
5
3
↑
5
3
2
2
3
↓
2
3
↑
5
4
3
2
1
↓
↑
↑
↓
↑
↑
Supported:
1
3
5
3
2
3