Algorithm

N patrons are independent threads, who behave identically, following some predefined rules.
theaterMan is the in charge of organizing the SHOW or maintaining the theater.
number of Seats in the theater is a Semaphore required.

theaterMan...
    waits until theater is full
    then starts the show and
    after the show is finished,
    waits until theater is empty.
    ...
    repeats above steps T times.

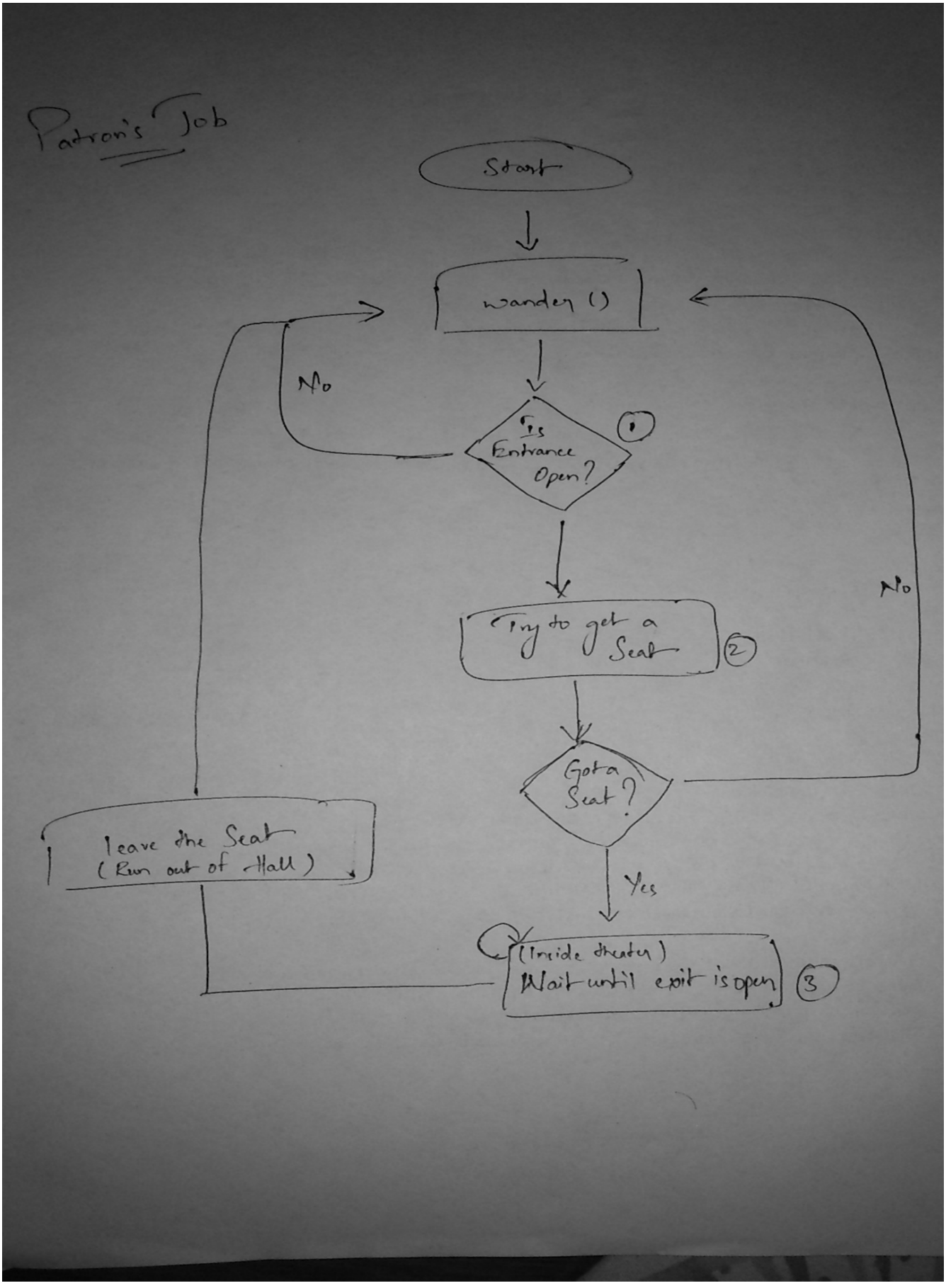
Two mutexes guard the Entrance and Exit, to stop any patron from entering or exiting the hall respectively in non appropriate times.

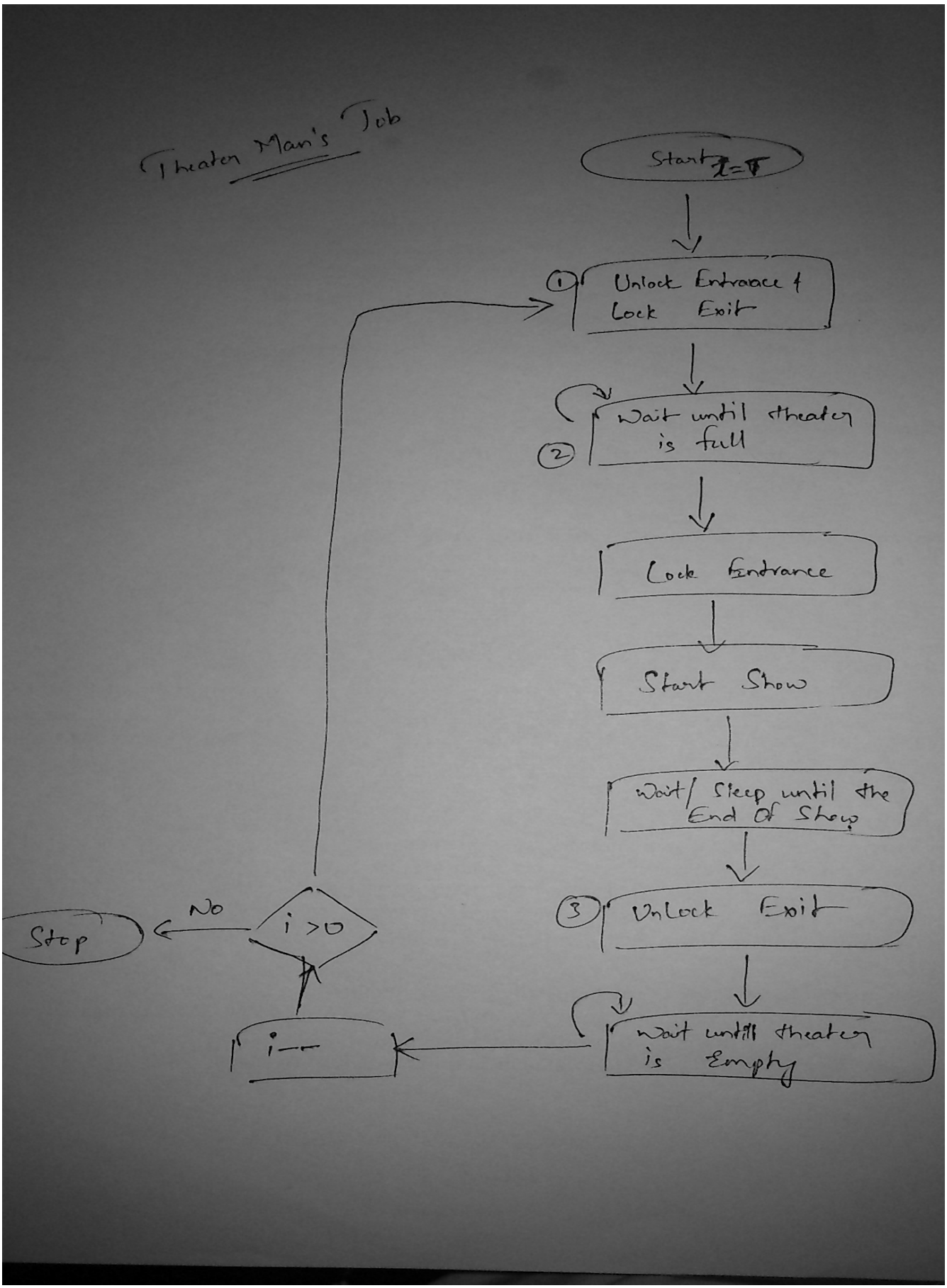
Exiting can also be achieved using an Exit flag, that will be set by theaterMan and read by patrons.
This will lead to busy waiting by patrons as they have to check for flag as long as the show is running.
patrons can not even sleep for show time, (they aren't kids to sleep during show) as logically patrons do not control show time.
Hence we need a mutex for Exit.

Until every patron in hall exits, no new patron should be allowed into the Hall.
A flag bit is required to specify valid times for entry.
    Which is again achieved using one more mutex Entrance.

patron's waiting time is reduced wherever possible, by implementing try-waits instead of waits.

More clear job descriptions of everyone are presented in the flow charts below.
    States that are marked with numbers inside circles are dependent on same numbered mark of other threads.


Patron's Job
Start
wander ()
No
Is Entrance Open?
1
Try to get a Seat
2
No
Got a Seat?
Yes
leave the Seat (Run out of Hall)
(Inside theater) Wait until exit is open
3


Theater Man's Job
Start i=T
1
Unlock Entrance & Lock Exit
2
Wait until theater is full
Lock Entrance
Start Show
Wait/ Sleep until the End Of Show
3
UnLock Exit
Wait until theater is Empty
i--
i >0
No
Stop