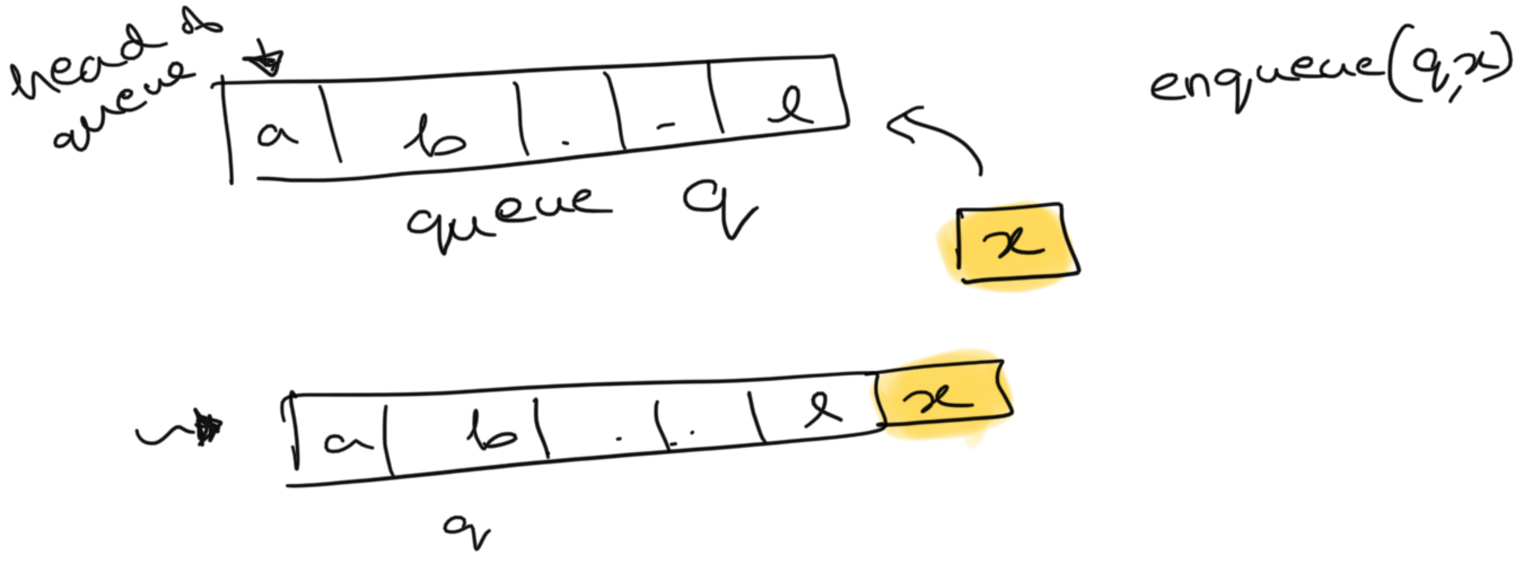
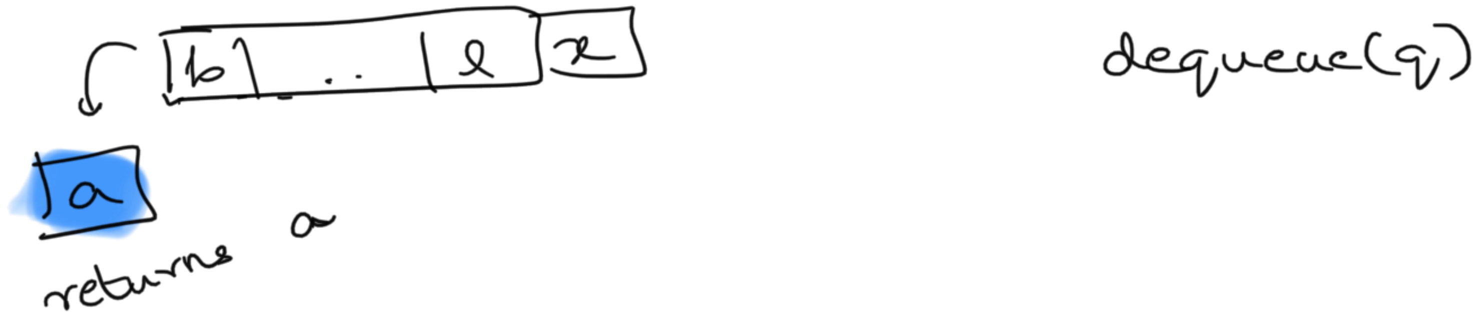
Queues

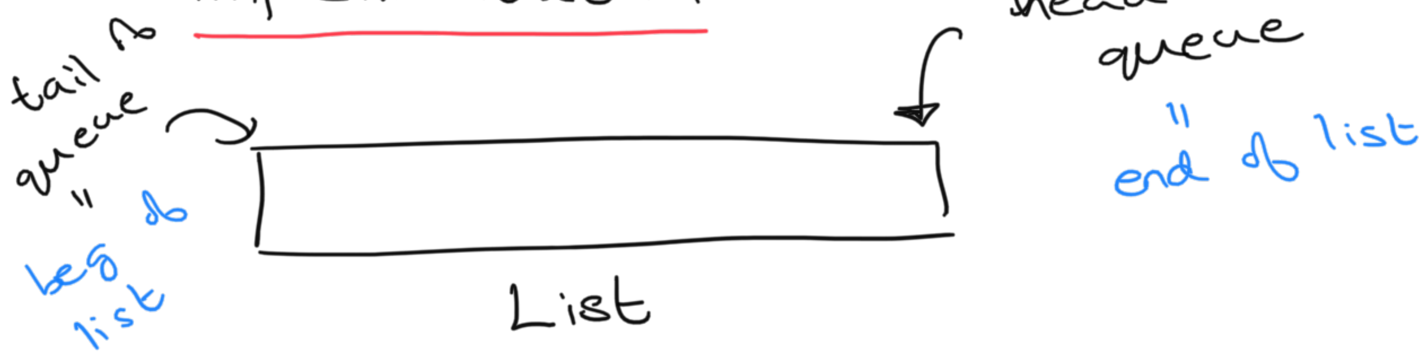
What is it?

First in, first out





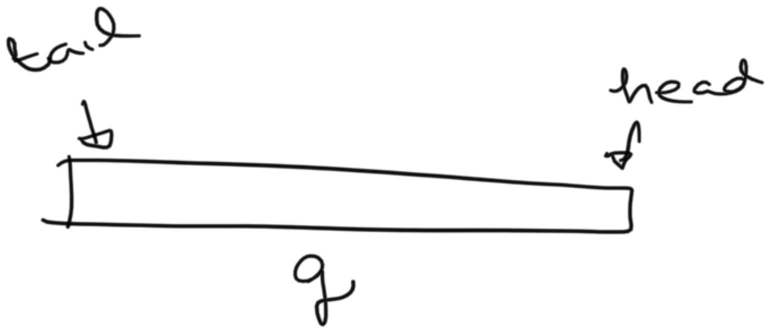
Implementation



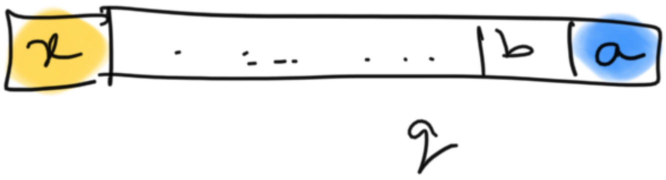
list has an insert method
l

l. insert (j,x) inserts x before position j

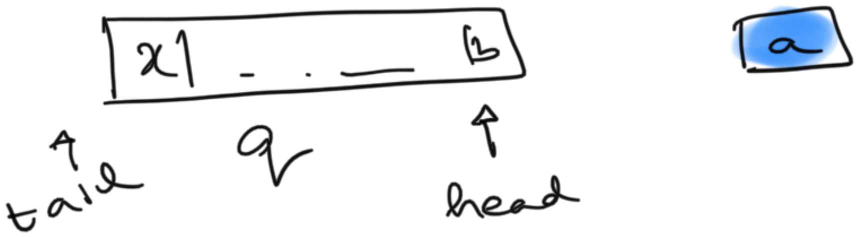
So l. insert (0,x) inserts x in beg. of list



enqueue (q, x) ⟷ q.insert(0, x)



dequeue (q) ⟷ q.pop ()



Used for

- systematic backtracking (bfs...)

Example : Giving chessboard dim,
start pos. of a "knight",
target pos

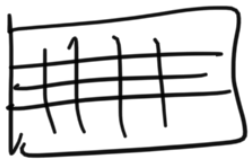
test if knight can travel from start to target.

Q

M

method : maintain queue, of positions to explore

marked or not



chess board array

I Initially

$Q = \boxed{\text{start pos}}$

M ← all positions on chess board unmarked. except start.

II (a) dequeue(Q) ⇝ "pos"

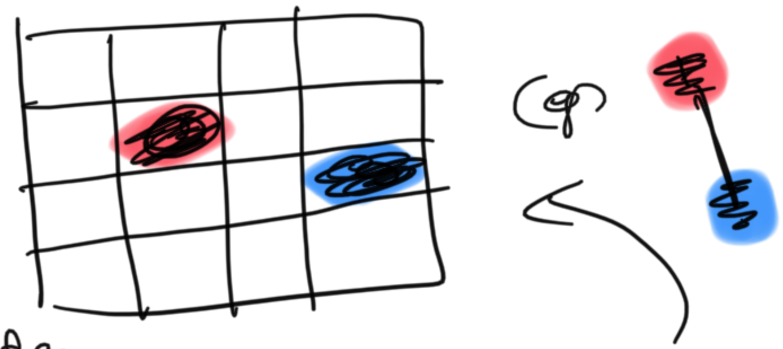
look at all positions which are unmarked and reachable by 1 knight move, and enqueue them

(b) mark them also in M

III Repeat step II till you reach target / Q empty..

BFS

- each square is a node.



- connect 2 nodes by 1 edge can be reached by 1 knight's move

⇝ graph

⇝ do bfs from start pos to see if target reachable.