Towers Of Hanoi

3 needles: src, aux, dest.

64 disks of decreasing sizes. stacked on the src needle.
↳ variable. (n)

Goal: Move all disks from src to dest
(preserve the order)

Rules: ① only one disk can be moved at a time.

② A larger disk can not be stacked above a smaller disk.

How many (minimum) number of moves needed?

$O(2^n)$

Case I: $n = 1$ $f(n) = 1$ (1 step)

Case II: $n = 2$ $f(n) = 3$ (3 steps)

Case III: $n = 3$ $f(n) = 7$

$$f(n) = 2^n - 1$$

n = 3 steps.

① Move 2 disks from src to aux.
② Move 1 disk from src to dest.
③ Move 2 disks from aux to dest.

General N:

① Move N-1 disks from src to aux.
② Move 1 disk from src to dest
③ Move N-1 disks " aux to dest.

$f(N) = 2 f(N-1) + 1$

Assume that $f(N-1) = 2^{N-1} - 1$

Then, $f(N) = 2(2^{N-1} - 1) + 1$

$= 2(2^{N-1} - 1) + 1$

$= 2^N - 2 + 1 = 2^N - 1$

Recursive Sol^n: - easier to code
- also efficient [$O(2^N)$ = # steps]

Algo. Design Towers. arguments.

1). # disks.

2) src.

3) dest

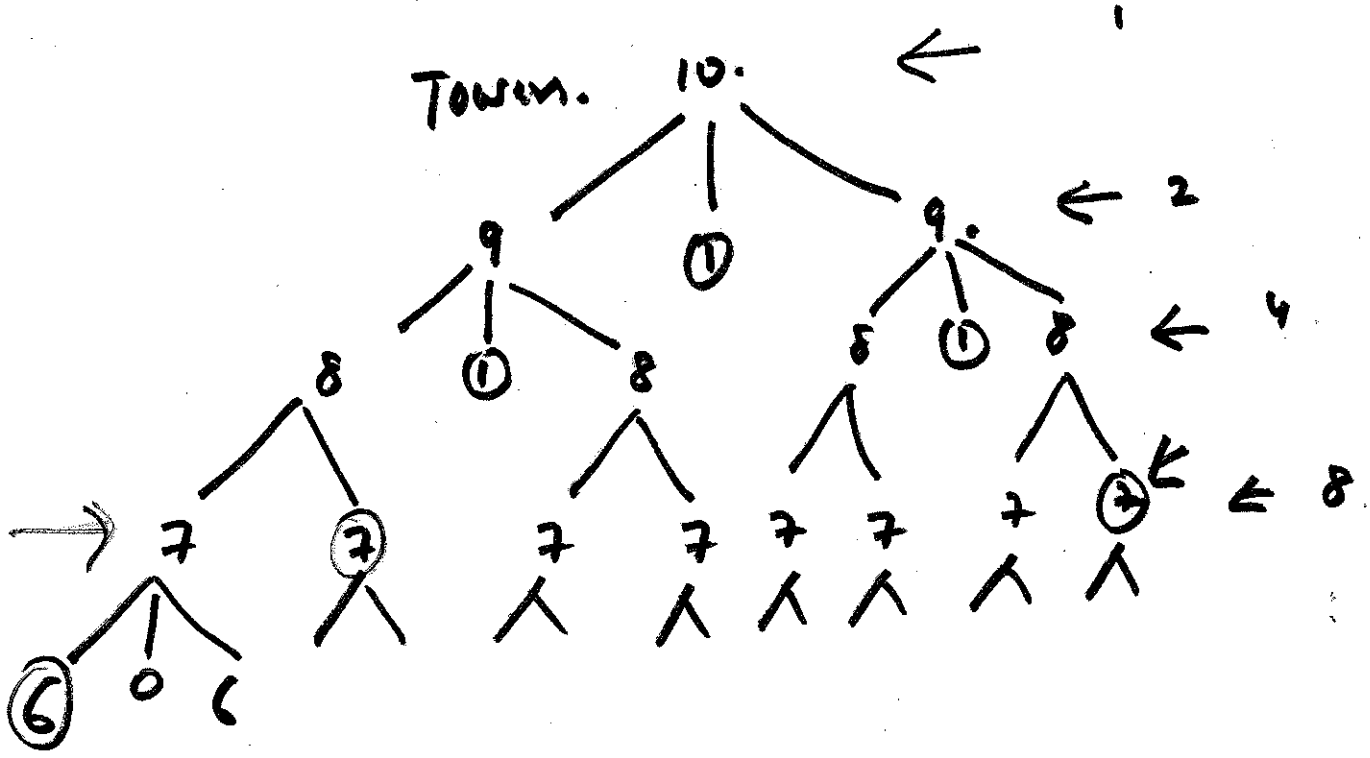
4) aux.

Towers (N, src, dest, aux)

① Towers (N-1, src, aux, dest)

② Move 1 disk from src to dest. [Base Case]
[Towers (1, src, dest, aux)] ← X.

③ Towers (N-1, aux, dest, src)



Stacks. (Ch. 3)

Linear lists.

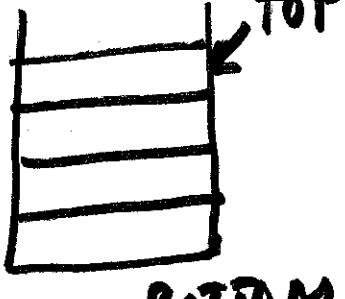
In a linear list, each element has a unique successor.

General list - data (element) can be inserted or deleted anywhere in the list.

Restricted list - data can only be inserted or deleted at the ends of the list.

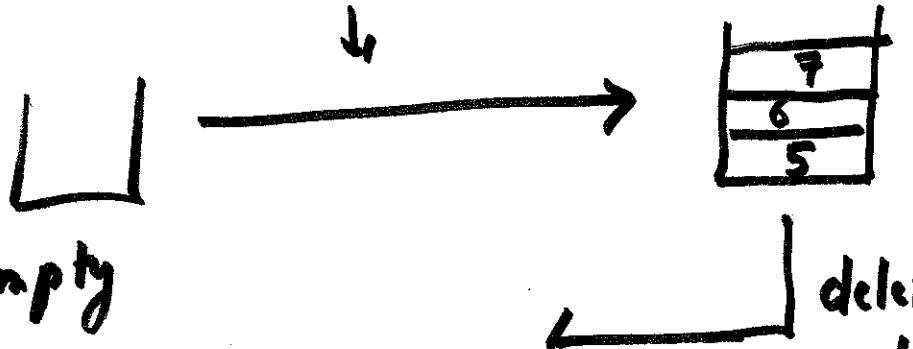
Queues. (FIFO)

Stacks (LIFO)



Insert & delete allowed from the TOP only.

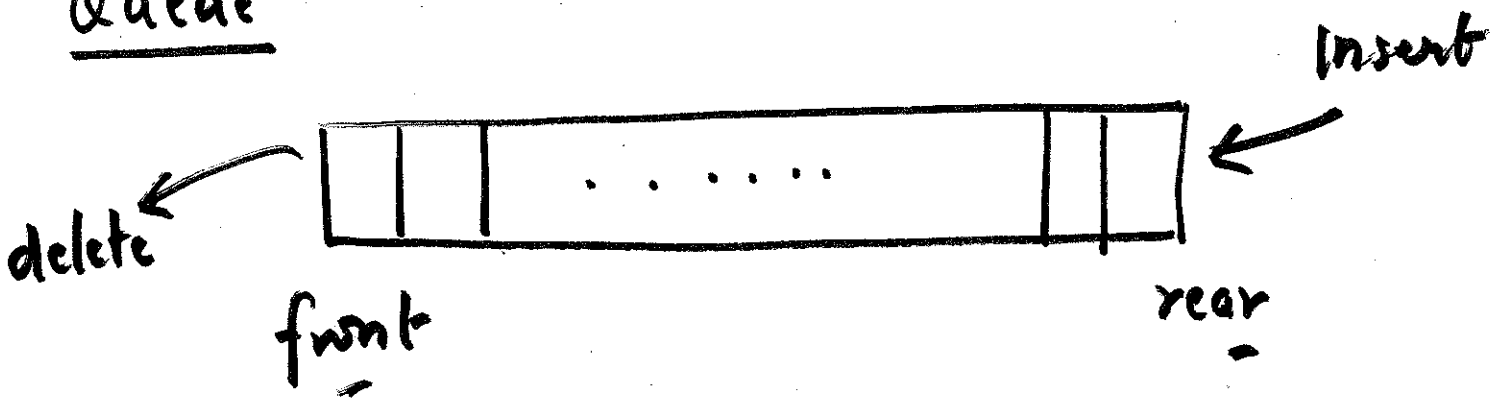
Insert 5, then 6, then 7.



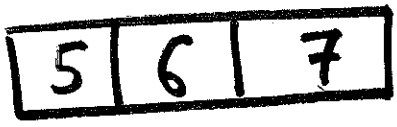
7 then 6 then 5.

(LIFO)

Queue



Insert 5, then 6, then 7



FIFO

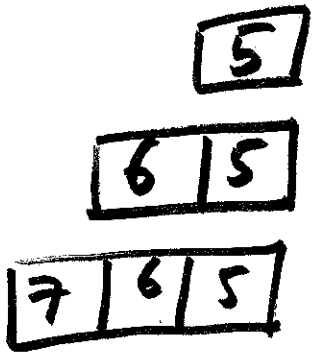
delete all 3.

5 then 6 then 7 (Queue Empty)

DeQueue. (Double Ended Queue)

↳ allows insertion and deletion at both ends.

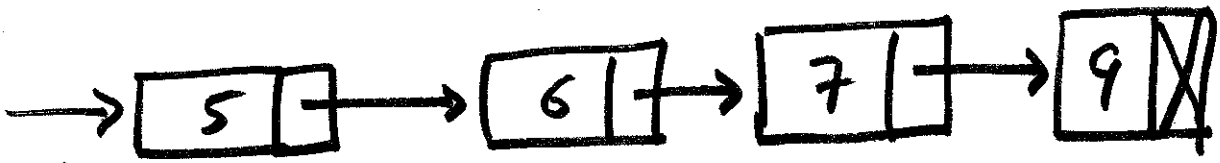
Insert 5, then 6, then 7.



delete all 3

↳ 5 7 6 (feasible)

General List:



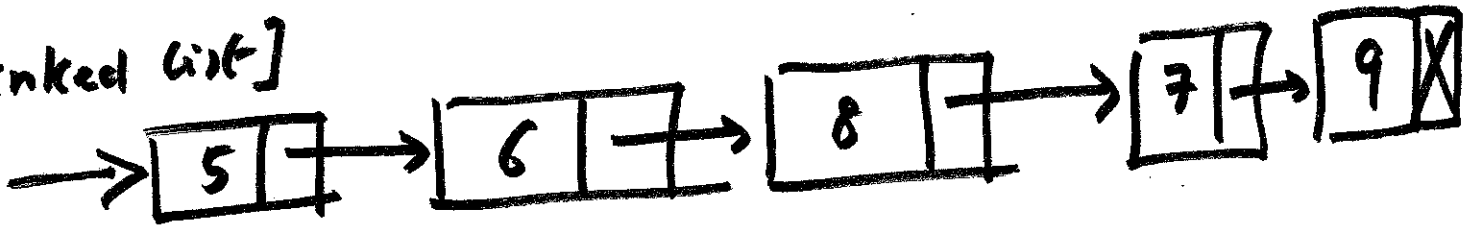
head (front)

(rear)

Insert & delete possible from anywhere in the list.

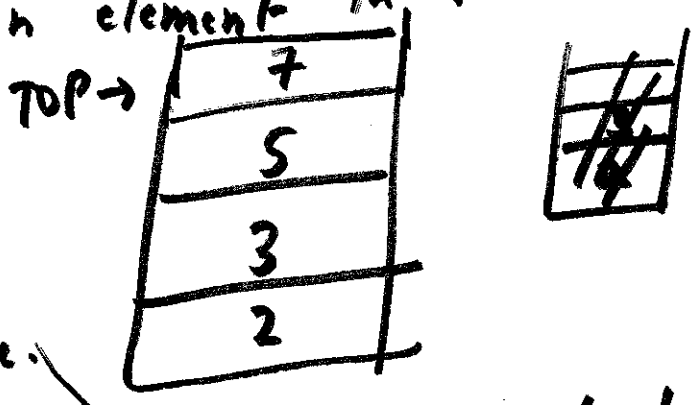
Insert 8 betⁿ 6 and 7

[Linked list]



Stacks:

Q. How to remove an element in the middle!



Remove 3.

delete → 7 → temp.

delete → 5 → store.

delete → 3 → use it.

→ insert back on stack.

Basic Operations:

PUSH, POP., Stack TOP, ...