

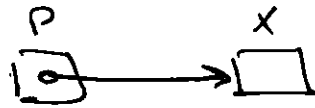
- Pat ext. to Tues. 10/13
- Quiz 1 on Tue 10/13

Queue Example

Stuff you should know

- memory commands
malloc, calloc, realloc
free

- Pointers



address-of op. $\&$

$$P = \&X$$

value-at op. $*$

$*P$ evaluates to X

typedef stmt :

typedef A B ;

↑
alias
for

↑

To make type called NodeObj & Node

typedef struct NodeObj {

} NodeObj ;

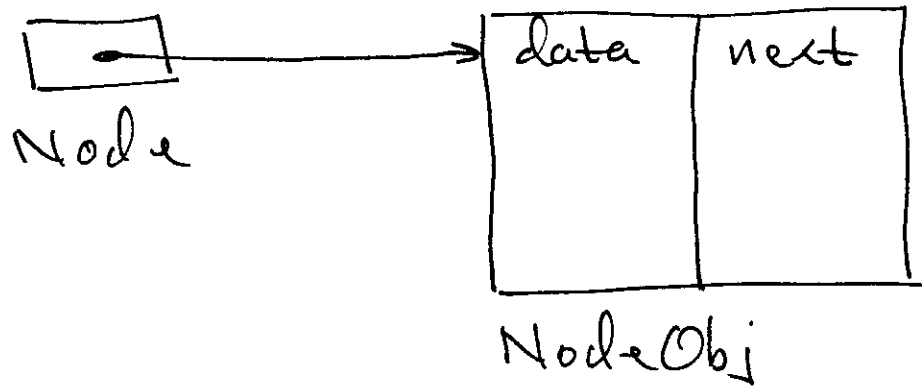
↑
B

↑
A

typedef NodeObj* Node ;

↑
A

↑
B



to make Queue & QueueObj

typedef struct QueueObj* Queue; ← h file

↑
A

↑
B

typedef struct QueueObj { ← .c file

...

} QueueObj;

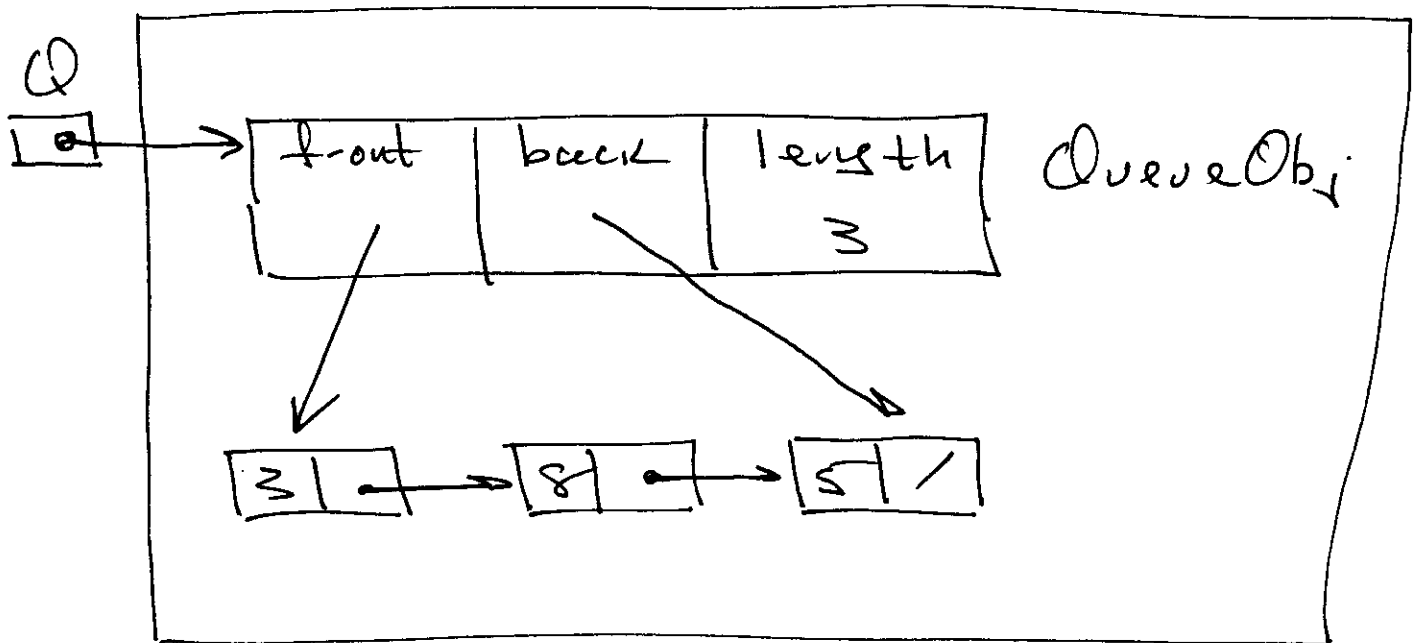
↑
B

↑
A

client view:

(3, 8, 5)

inside-the-box ADT view:

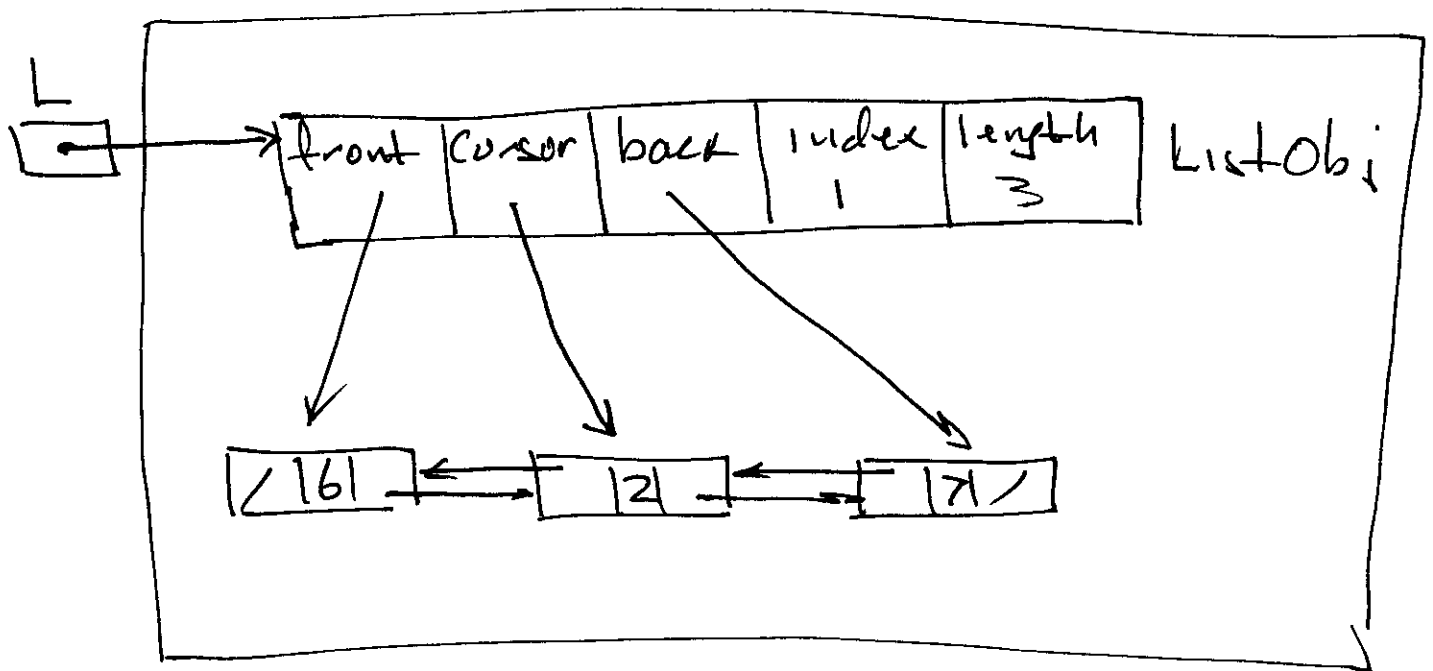


Part: List ADT

client view:

$$\begin{matrix} & 0 & & 1 & & 2 \\ [& 6 & , & 2 & , & 7 &] \end{matrix}$$

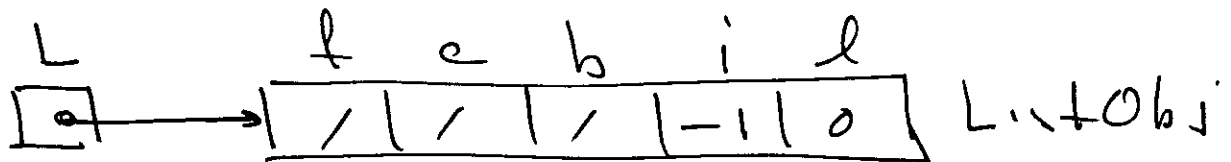
inside view:



empty state:

client : ()

inside:



not empty state:



this is a NULL List reference, not a List in the empty state.

How to indirectly sort an array
using only List ops.

Ex.

$$A = \begin{array}{|c|c|c|c|} \hline \overset{0}{\cancel{0}} & \overset{1}{\cancel{1}} & \overset{2}{\cancel{2}} & \overset{3}{3} \\ \hline "c" & "a" & "b" & "d" \\ \hline \end{array}$$

want $L = (1, 2, 0, 3)$ ←

start: $L = ()$

insert 0: $L = (0)$

insert 1: $L = (\underline{0})$

$L = (1 \underline{0})$

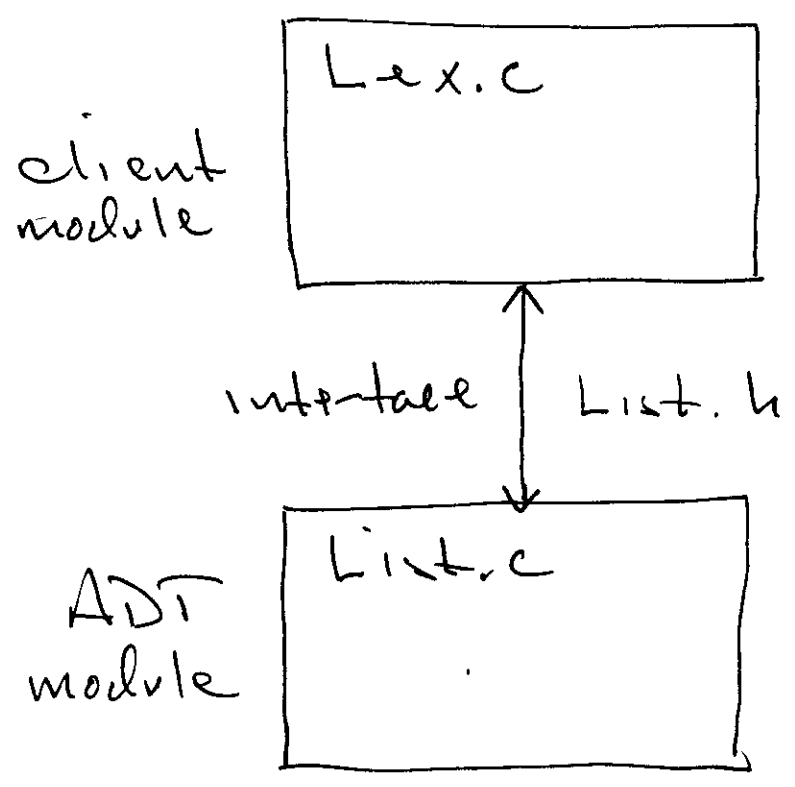
insert 2: $L = (\underline{1} \underline{0})$

$L = (1 \ 2 \ \underline{0})$

insert 3: $L = (\underline{1} \ \underline{2} \ \underline{0})$

$L = (1 \ 2 \ 0 \ 3)$ ←

structure chart



ADTs vs. Data Structures.

