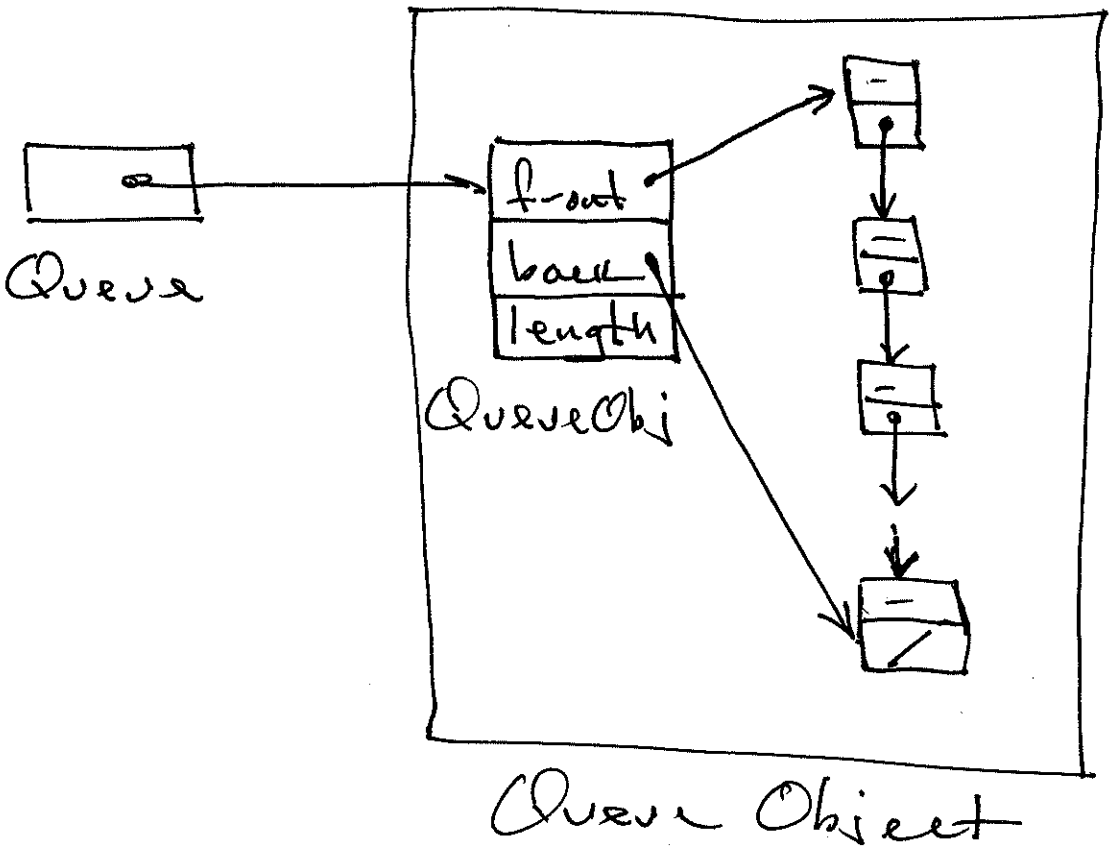
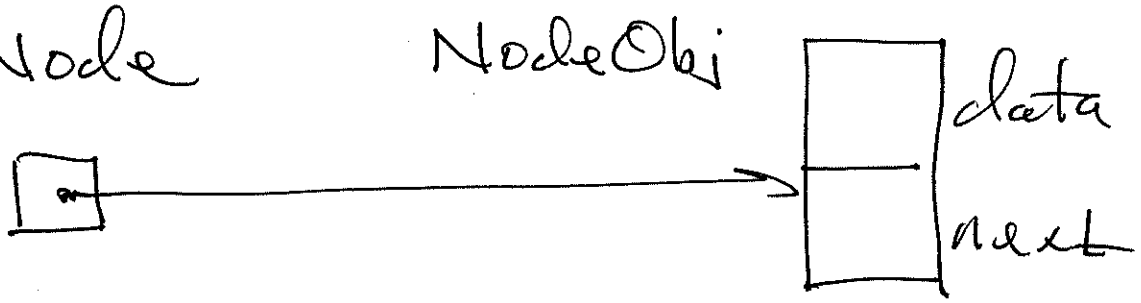
CSC 101-02 3-31-22

Recall Queue example



Private type:

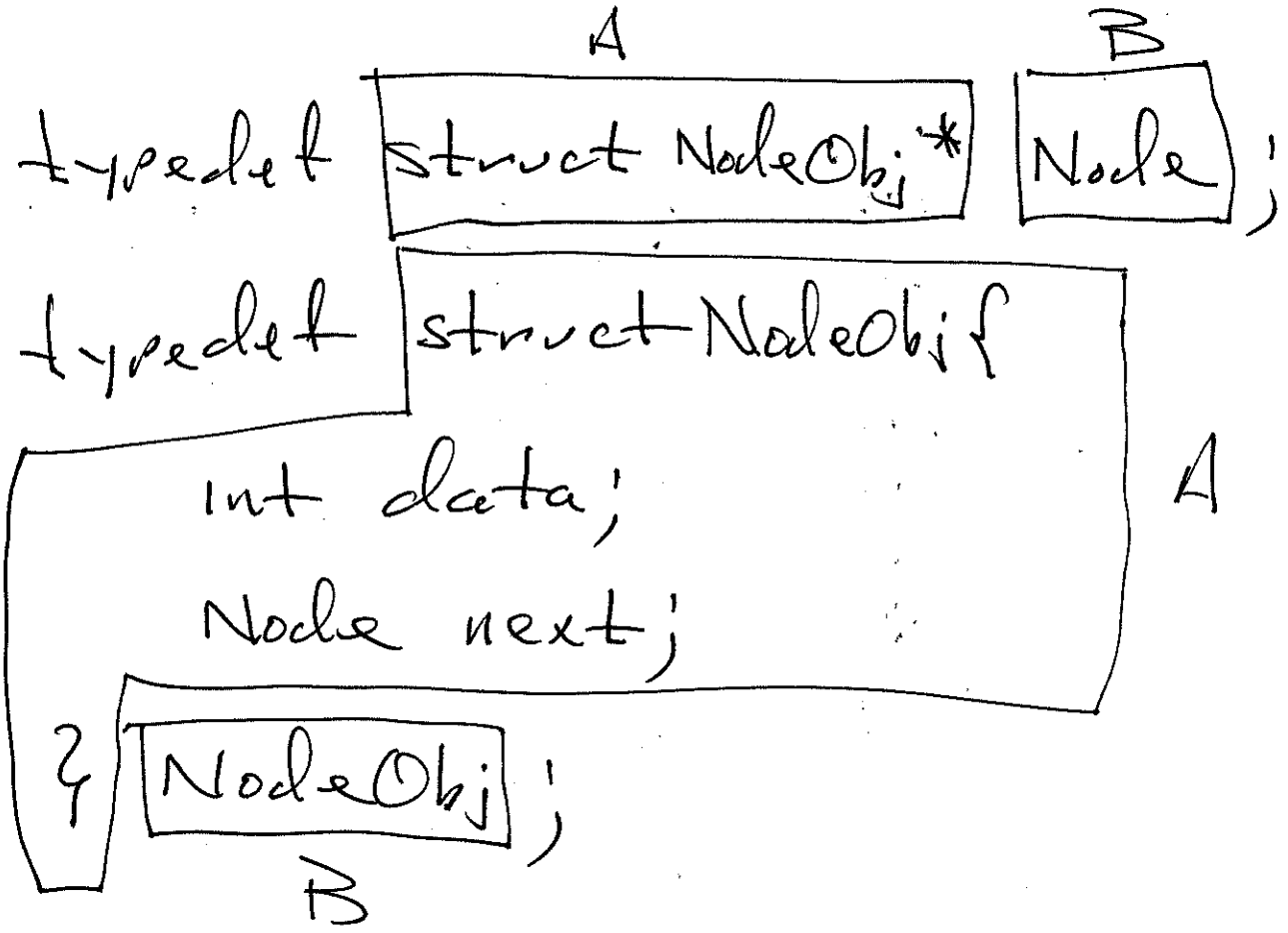
Node NodeObj



typedef A B;

↑

alias for A



See Queue.h

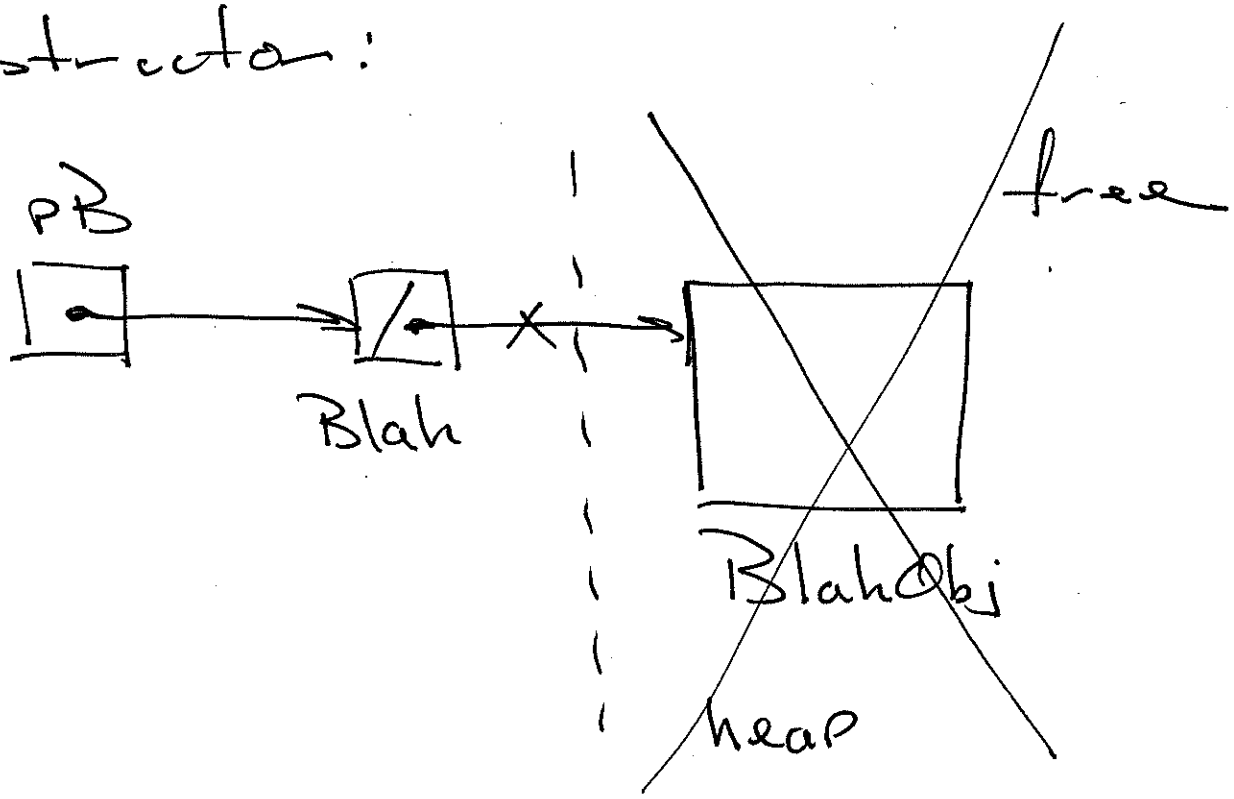
Queue.C

QueueTest.c

note:

(*N).data same as N->data

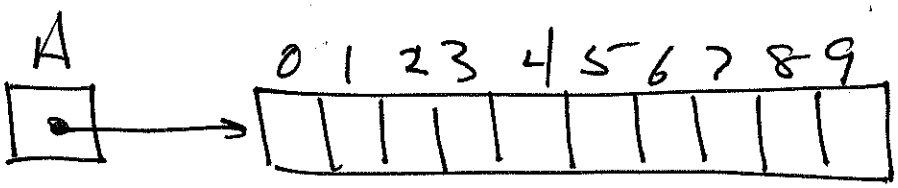
Destructor:



Variable length Arrays in C (VLA)

To create an array do:

```
int A[10];
```



all stack memory

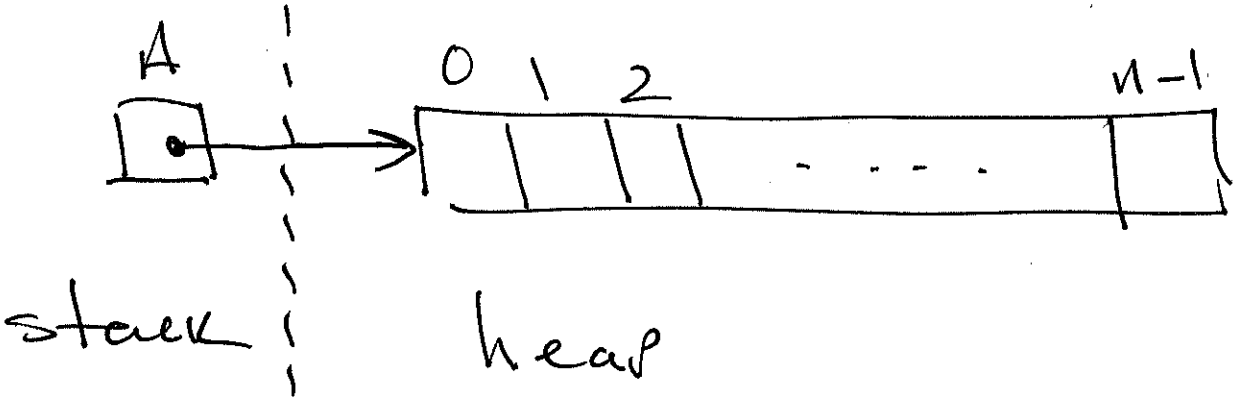
Can also do (since ISO c99):

```
int n; // get n from user
int A[n];   <-- Don't do this
for (i=0; i<n; i++){
    A[i] = something;
}
```

Don't use VLA's

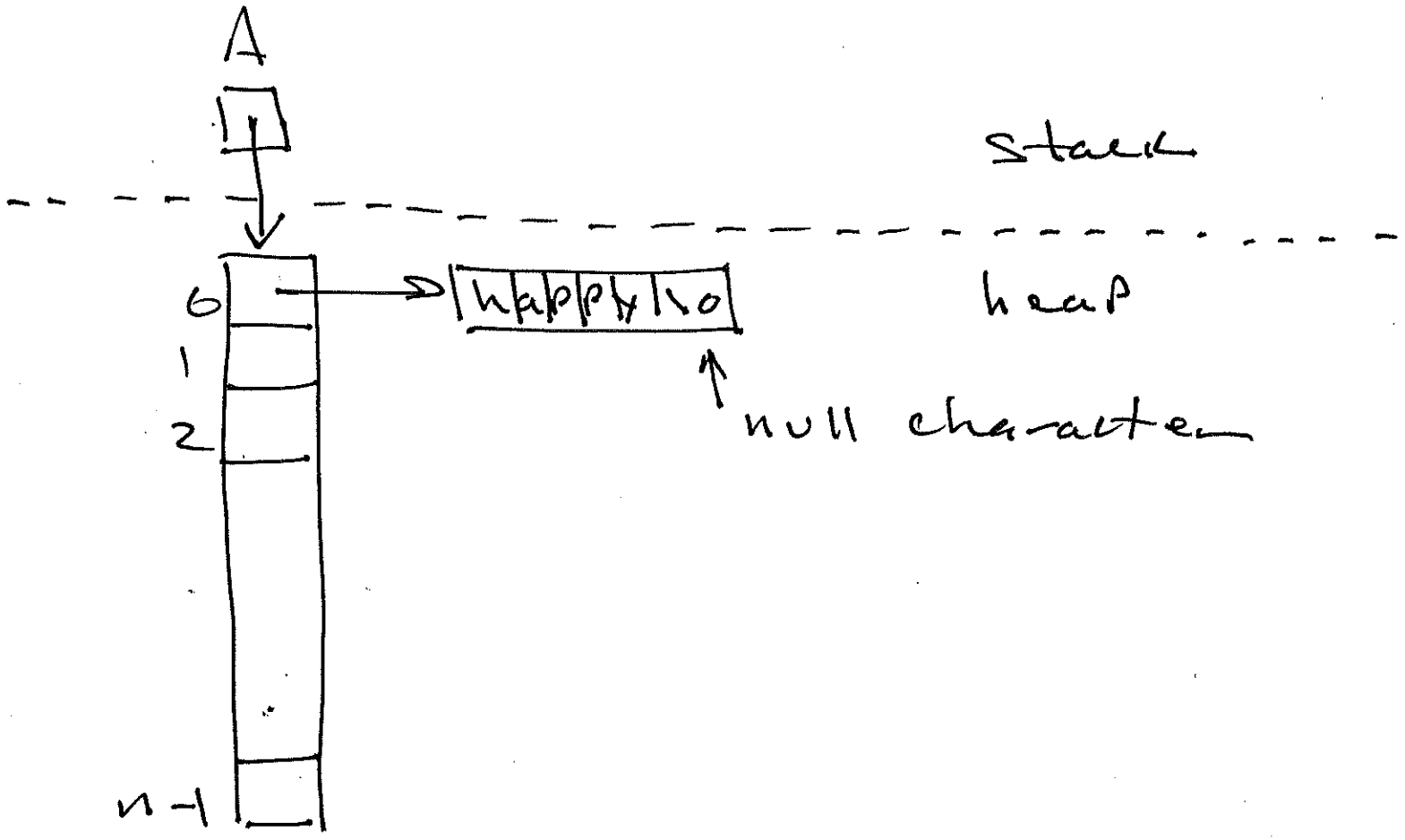
instead use heap memory

```
int n; // get n somehow.
int* A;

A = calloc(n, sizeof(int));
```




```
// later

free(A);
A = NULL;
```

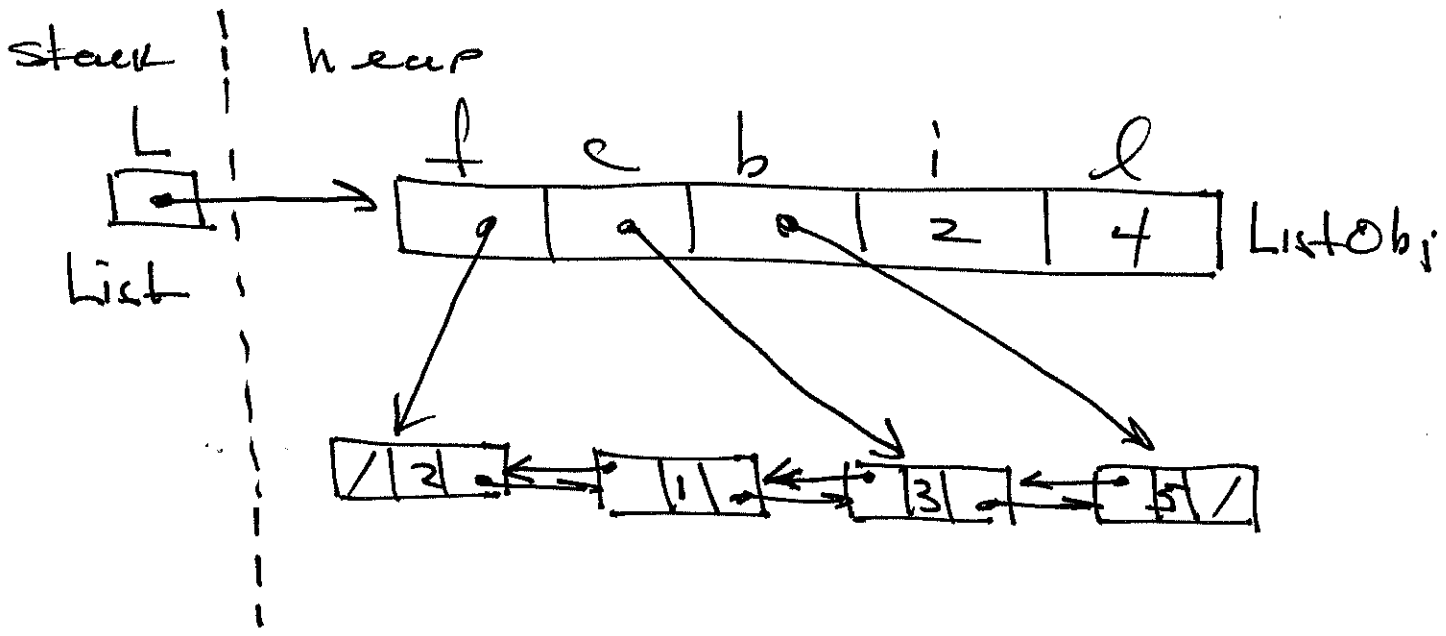
Pal : Lex.c



List ADT

client view: (2, 1, 3, 5) (0, 1, 2, 3)

inside view:

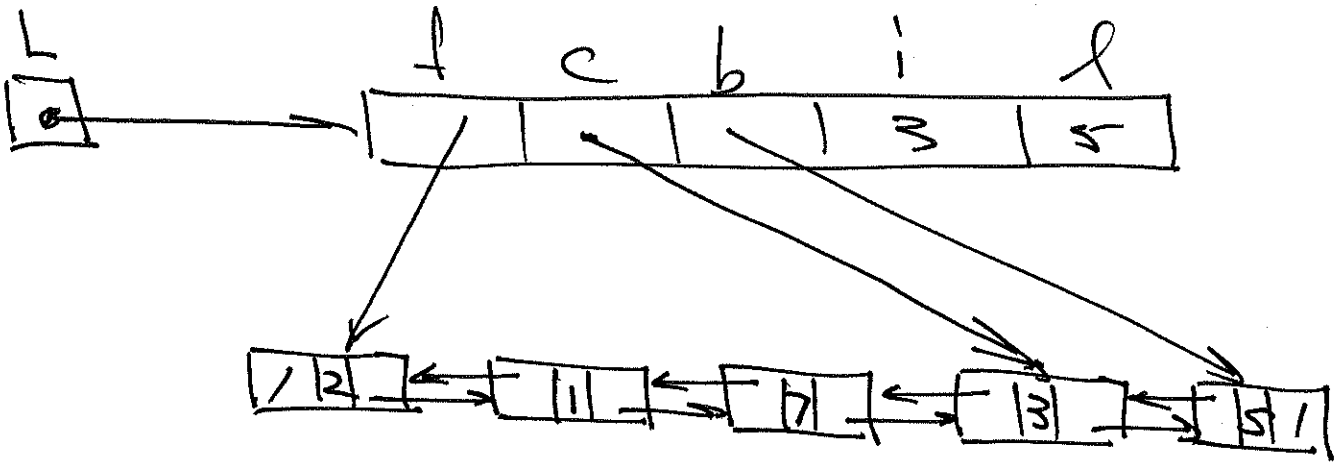


Do: InsertBefore(L, 7)

0 1 2 3 4

client view: (2, 1, 7, 3, 5)

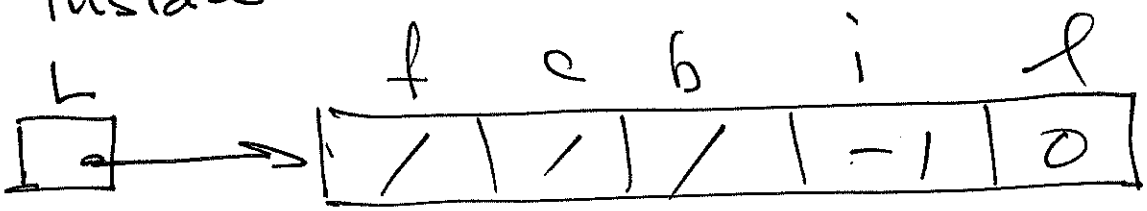
inside view:



Empty state:

client: ( )

inside



Example: suppose List ADT is done.
how to insert array indices into list.

$$A = (\underset{0}{\text{"c"}}, \underset{1}{\text{"a"}}, \underset{2}{\text{"b"}}, \underset{3}{\text{"d"}})$$

Want: $L = (1, 2, 0, 3)$

start: $L = ( )$

insert 0: $L = (0)$

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

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

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

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

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

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

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

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

$L = (1, 2, 0)$

$L = (1, 2, 0, 3)$