

A List (Math object)

a sequence of zero or more elements of the same type.

$$\langle a_0, a_1, a_2, a_3, \dots, a_n \rangle$$

where a_i are the elements of the list.

if $n < 0$, the list is empty $\langle \rangle$

$n+1$ is the length of the list.



elements in a list are linearly ordered:

Every element (except the last one a_n) has a successor.

Every element (except the first one a_0) has a predecessor.

• element a_i is said to be at position i

[Operations

size(L)

first(L)

last(L)

at(L, i)

find(L, x): true if $x \in L$
false otherwise

$$L_1 = \langle a, r, n, s, l \rangle$$

$$\text{size}(L_1) = 5$$

$$\text{first}(L_1) = a$$

$$\text{last}(L_1) = l$$

$$\text{at}(L_1, 3) = s$$

$$\text{find}(L_1, r) = \text{false}$$

$$\text{find}(L_1, s) = \text{true}$$

insert(L, x, i): insert x in L
at position i

insert(L1, c, 3) = (a, r, n, c, s, l)

remove(L, i): remove object at
position i

remove(L1, 2) = (a, r, s, l)

There is no fixed set of operations.

reporters & transformers
size, front take a list return a list

simple & complex
described english algorithmically in terms of simple functions

swap(L, i, j):

pre: objects a_i and a_j exist in L

$X \leftarrow \text{at}(L, i)$

$L \leftarrow \text{insert}(L, j, X)$

$Y \leftarrow \text{at}(L, j+1)$

$L \leftarrow \text{remove}(L, j+1)$

$L \leftarrow \text{remove}(L, i)$

$i < j$

Data Structure for List?

Implementation: Class

- members = Array
- operations = member functions.

= the
Array List
class