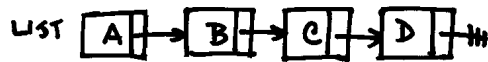


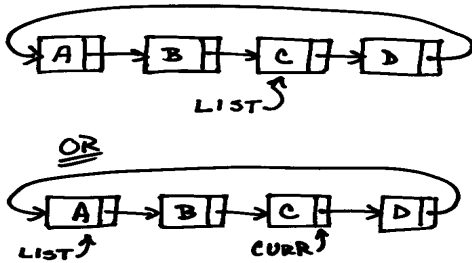
List Variations

- Linear Linked List
- Circular Lists
- List with header nodes
- Doubly-linked list
- Multi-linked List



List Variations: Circular

- Useful to process any item by going forward
- Can reach items behind current location without starting over



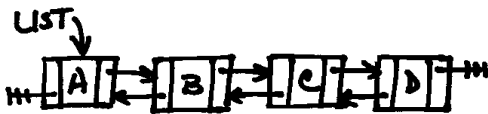
List Variations: Header nodes

- Useful for working with lists that are frequently empty



List Variations: Doubly-linked lists

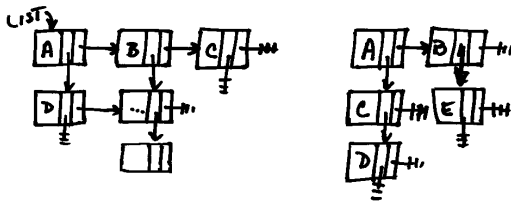
- Useful for changing direction in list frequently
- Useful for frequent deletions



35

List Variations: Multi-linked Lists

- Useful to follow more than one kind of path through data
- Can become Inverted or Threaded file
 - (see Horowitz and Sahni)



List Examples

- Josephus problem
 - Once you start, current position is all that matters.
- Very long Integers
- Polynomial Arithmetic
- Sparse Matrices

37

JOSEPHUS PROBLEM

EXAMPLE:

$N = 3$
A, B, C, D, E are five soldiers
A was picked from hat

D WILL VOLUNTEER

. 8

List Examples: Very Long Integers

Motivation: Public Key/Private Key Encryption

Person A

MSG

CODE WITH A's PRIVATE KEY

CODE WITH B's PUBLIC KEY

UNSECURE TRANSMISSION

DECODE w/ A's Public Key

DECODE w/ B's Private Key

DECODE Person B

39

List Example: Very Long Integers

999423621587
+ 759841235
1000183462822

NUM1

NUM2

SUM

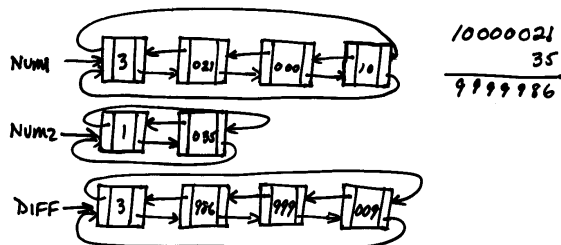
List Examples: Very Long Integers

- Use pointers in each list to move through them in parallel.
- Use of header keeps track of original start of list.
- Note size of sum is different from size of addends.

41

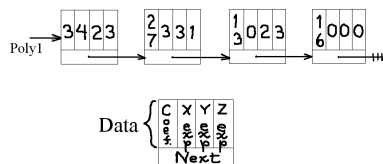
List Examples: Very Long Integers

- Use double linked list to support subtraction. Why?



List Examples: Polynomial arithmetic

$$3x^4y^2z^3 + 27x^3y^3z + 13y^2z^3 + 16$$



43

List Variations: Multi-linked Lists

$$\begin{bmatrix} 0 & 0 & -2 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 2 \\ 0 & 7 & 0 & 0 & 0 \end{bmatrix}$$
