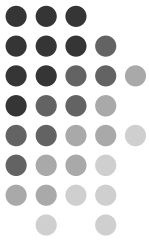
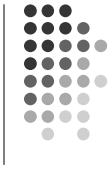
Data Organization:
Creating Order Out Of Chaos

3A

Fundamental Data Structures



Data Structures



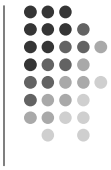
- The organization of data is a very important issue for computer scientists.
- A **data structure** is a way of storing data in a computer so that it can be used efficiently.
  - Choosing the right data structure will allow computer scientists to develop certain algorithms for that data that are more efficient.
  - A **vector**, or **one-dimensional array**, is a very simple data structure for holding a sequence of data.

A Pointer

Linked Lists

- Another data structure that stores a sequence of data values is the **linked list**.
- Data values in a linked list do not have to be stored in adjacent memory cells like a vector.
- To accommodate this feature, each data value has an additional “pointer” that indicates where the next data value is in computer memory.
- In order to use the linked list, we only need to know where the first data value is stored.

Linked List Example



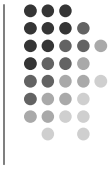
- Linked list to store the sequence: [4, 9, 17, 23, 29]

Assume each data value and pointer requires 2 bytes.

List starts here →

Memory Cell Address	Data Value	Pointer (next)
100	9	124
104	29	0 (null)
108		
112	4	100
116	23	104
120		
124	17	116
128		

Inserting a new data value



- One-dimensional Array (Vector)

	1	2	3	4	5	6	7	8	9	10
A	4	9	17	23	29					

- We have to move potentially the entire set of numbers over to make room for the new value, depending on where we want to insert it. (Example: Insert 12)

	1	2	3	4	5	6	7	8	9	10
A	4	9	12	17	23	29				

Inserting a new data value

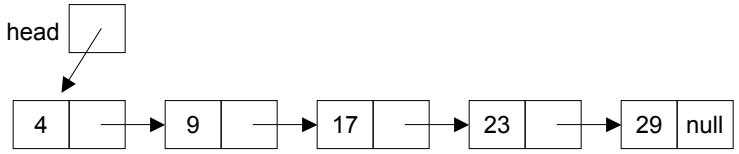
- Linked List
  (Insert 12)

List starts here →

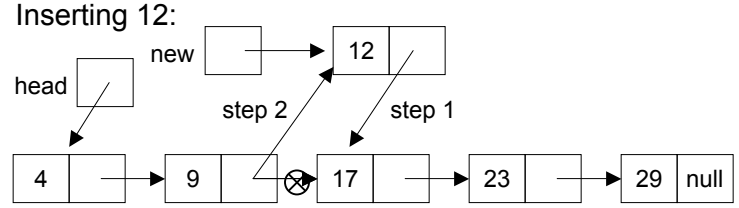
Memory Cell Address	Data Value	Pointer (next)
100	9	128
104	29	0 (null)
108		
112	4	100
116	23	104
120		
124	17	116
128	12	124

Drawing Linked Lists

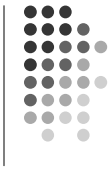
- L = [4, 9, 17, 23, 29]



- Inserting 12:



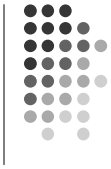
Comparison



- What are the advantages of using vectors?

- What are the advantages of using linked lists?

Tables



- Some data can be organized efficiently in a **table** (also called a **matrix** or **2-dimensional array**)
- Each cell is denoted with two subscripts, a row and column indicator

B[3,4] = 50

B	1	2	3	4	5
1	3	18	43	49	65
2	14	30	32	53	75
3	9	28	38	50	73
4	10	24	37	58	62
5	7	19	40	46	66

Table Example

Algorithm to find the sum of all data values in a matrix.

Input B, a 5 X 5 matrix of integers

Set sum = 0

For each row r of B do the following:

    For each column c of B
    do the following:

        Add B[r,c] to sum

Output sum

r	c	sum
5	5	949

B	1	2	3	4	5
1	3	18	43	49	65
2	14	30	32	53	75
3	9	28	38	50	73
4	10	24	37	58	62
5	7	19	40	46	66

Stacks

Stacks

- A **stack** is a data structure that works on the principle of Last In First Out (LIFO).
  - LIFO: The last item put on the stack is the first item that can be taken off.
- Common stack operations:
  - Push – put a new element on to the stack
  - Pop – remove the top element from the stack
- A stack can be implemented using a vector or a linked list.
- Applications: calculators, compilers, programming

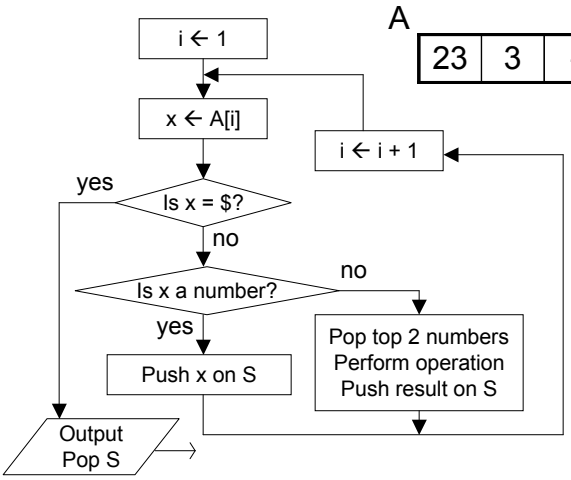
RPN

- Some modern calculators use Reverse Polish Notation (RPN)
  - Developed in 1920 by Jan Lukasiewicz
  - Computation of mathematical formulas can be done without using any parentheses
  - Example:
    (3 + 4) * 5 =
    becomes in RPN:
    3 4 + 5 *

Hewlett-Packard
12c financial calculator

Stack Example

Computing the value of an RPN string



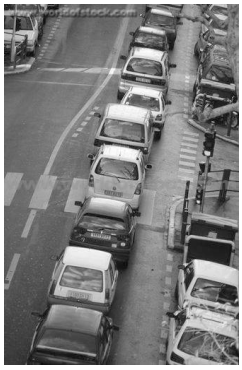
A

23	3	-	4	6	+	/	$

Stack Example

TRACE HERE

Queues

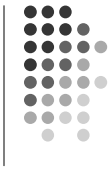


Queues

- A **queue** is a data structure that works on the principle of First In First Out (FIFO).
  - FIFO: The first item stored in the queue is the first item that can be taken out.
- Common stack operations:
  - Enqueue – put a new element in to the queue
  - Dequeue – remove the first element from the queue
- A queue can be implemented using a vector or a linked list.
- Applications: printers, simulations, networks

Queues

Josephus Problem



Given a group of **n** men arranged in a circle such that every **m**th man will be executed until one remains, find the position of the last survivor.

1. Fill Q with people numbered 1 through n.
2. While Q is not empty do the following:
    a. Do the following m-1 times:
        i. Let x = Dequeue Q
        ii. Enqueue x on Q
    b. Let x = Dequeue Q
3. Output x as the last survivor.

Queue Example

TRACE HERE n = 6, m = 4