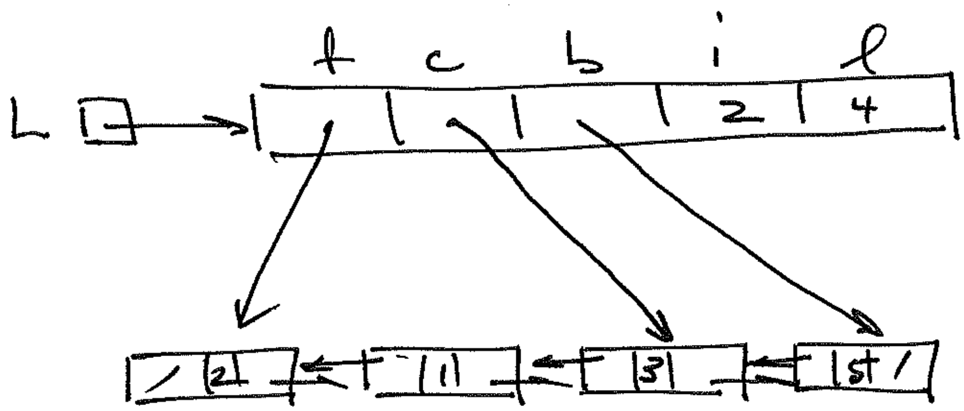
CSE 101 1-16-24

List ADT :

$$\begin{array}{lcccccc} & & 0 & 1 & 2 & 3 & \\ \text{client} : & (& 2 & 1 & \underline{3} & 5 &) \end{array}$$

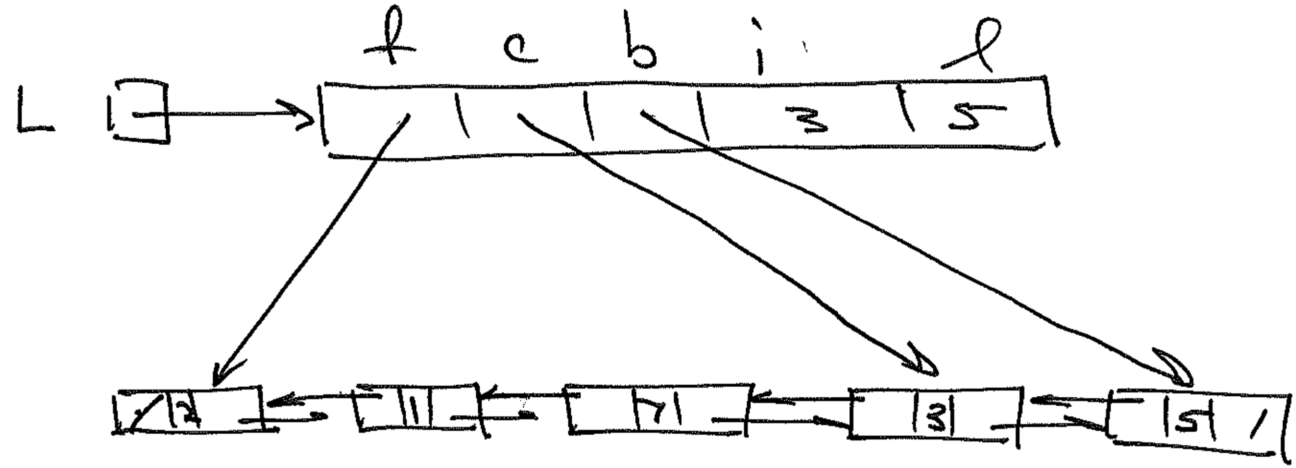
inside :



Do InsertBefore(L, 7)

$$\begin{array}{lccccccc} & & 0 & 1 & 2 & 3 & 4 & \\ \text{client} : & (& 2 & 1 & 7 & \underline{3} & 5 &) \end{array}$$

inside:



Pal Example

How to insert array indices into list

$$\begin{array}{cccc} 0 & 1 & 2 & 3 \end{array}$$

$A = [$ "c" "a" "b" "d" $]$

Goal

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

	Subarray
Start: $L = (\ )$	[ ]
insert 0: $L = (0)$	[c]
insert 1: $L = (\underline{0})$	
$L = (1 \ \underline{0})$	[a c]
insert 2: $L = (\underline{1} \ 0)$	
$L = (1 \ \underline{0})$	
$L = (1 \ 2 \ \underline{0})$	[a b c]
insert 3: $L = (\underline{1} \ 2 \ 0)$	
$L = (1 \ \underline{2} \ 0)$	
$L = (1 \ 2 \ \underline{0})$	
$L = (1 \ 2 \ 0)$	
$L = (1 \ 2 \ 0 \ 3)$	[a b c d]

note:

Don't use Variable length Arrays (VLAs) in C

as of ISO C99 this is Possible

```
int n;
// get n from user
int A[n];   // stack memory
// do something with A.
```

Don't do this!! use heap memory

```
int* A = calloc(n, sizeof(int));
   ⋮  // do something
   ⋮
free(A);
```

Graph Theory Handout

What is a graph?

$$G = (V, E)$$

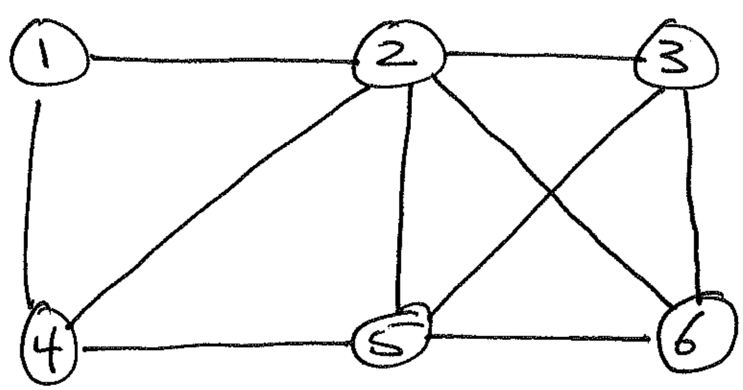
V: Vertex set $\neq \emptyset$

E: edge set

<u>note</u>: $12 = 21 = \{1, 2\}$

<u>Ex.</u> $V = \{1, 2, 3, 4, 5, 6\}$

$E = \{12, 14, 23, 24, 25, 26, 35, 36, 45, 56\}$



$\deg(6) = 3$

We say

- 1 is adjacent to 2
- 12 is adjacent to 23
- 2 is incident with 42
- 6 has degree 3
- neighbors of 6 are {3, 2, 5}

- a 1-6 <u>walk</u> of length = 6

  1, 4, 2, 5, 3, 2, 6 (also a <u>trail</u>)

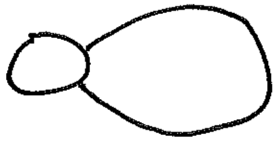
- a 1-6 <u>path</u> of length = 5

  1, 2, 4, 5, 3, 6

- a shortest 1-6 path (length = 2)

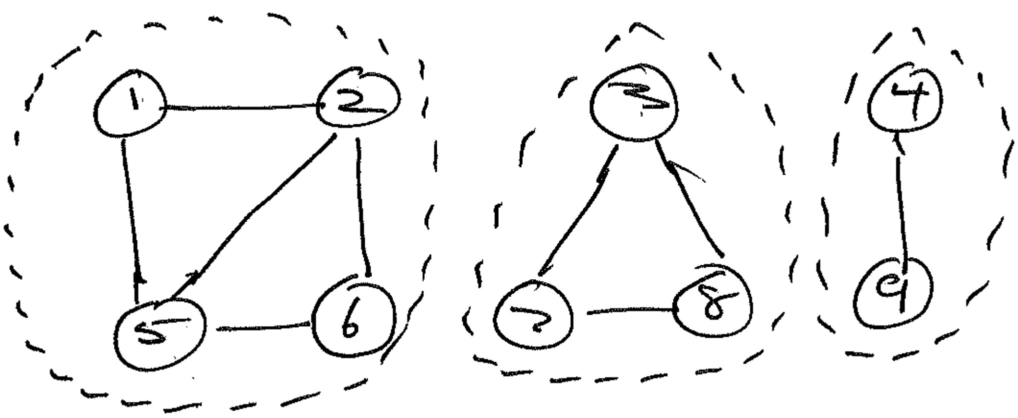
  1, 2, 6

<u>note</u>:

self loop  not allowed

Parallel edges  not allowed

<u>Ex</u>



<u>Defn</u> G is <u>connected</u> iff for all $x, y \in V(G)$, G contains an x-y path.

Defn

a subgraph H of G is called a connected component iff

(1) H is connected

(2) H is maximal w.r.t. (1)

Defn

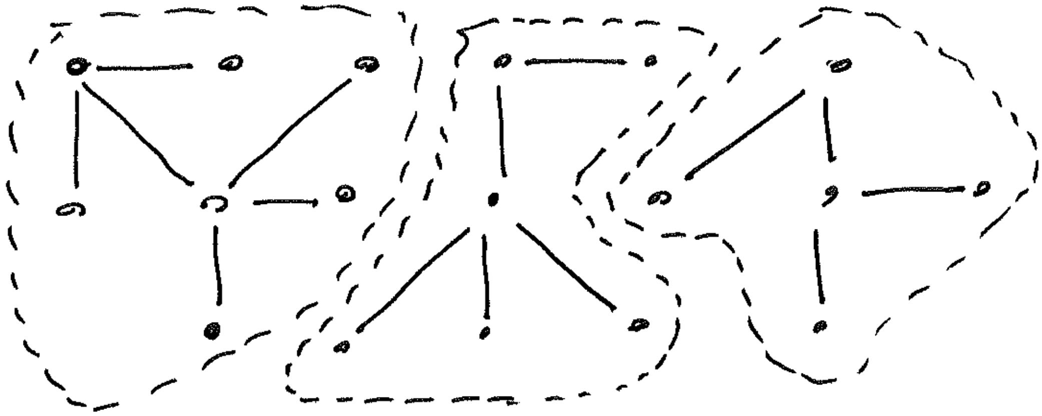
A cycle in G is a closed Path

Defn

A Tree is a graph that is both Acyclic (contains no cycles) and connected.

Defn An acyclic graph is called a forest.

Ex. forest:



#Vert:	7	6	5
#edges:	6	5	4

Read section on <u>directed</u> Graphs P. 7–8 in handout.

<u>next time</u>:

Representations of Graphs & Digraphs: P. 9–10.