

Estructuras de Datos

—

Ayudantía 3

Agenda

- Listas Ligadas
- Grafos

Listas Ligadas



Array

```
tipo[] nombre_variable = new tipo[num_elems]
```



memoria reservada = tamaño(tipo) x num_elems

¿Cómo agregamos
elementos cuando
ya no quedan
espacios?

```
if index > array.length:
    tipo[] new_array = new tipo[index+index/2]
    for i; i < array.length ; i++:
        new_array[i] = array[i]
    new_array[index] = item
    index += 1
    array = new_array
else:
    array[index] = item
```

Listas Dinámicas

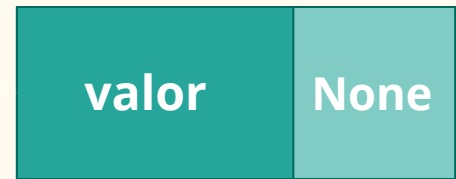
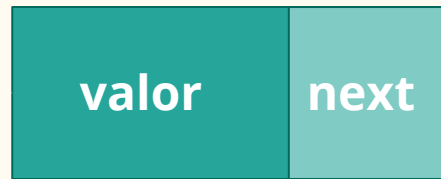
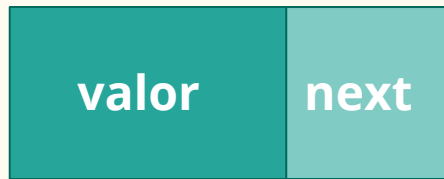
Ventajas

- Fáciles de usar
- Comportamiento encapsulado

Desventajas

- Gasto de memoria
- Desventajas de uso del array

Listas Ligadas



¿Por qué preferimos
una lista ligada a una
lista dinámica?

Eficiencia :)

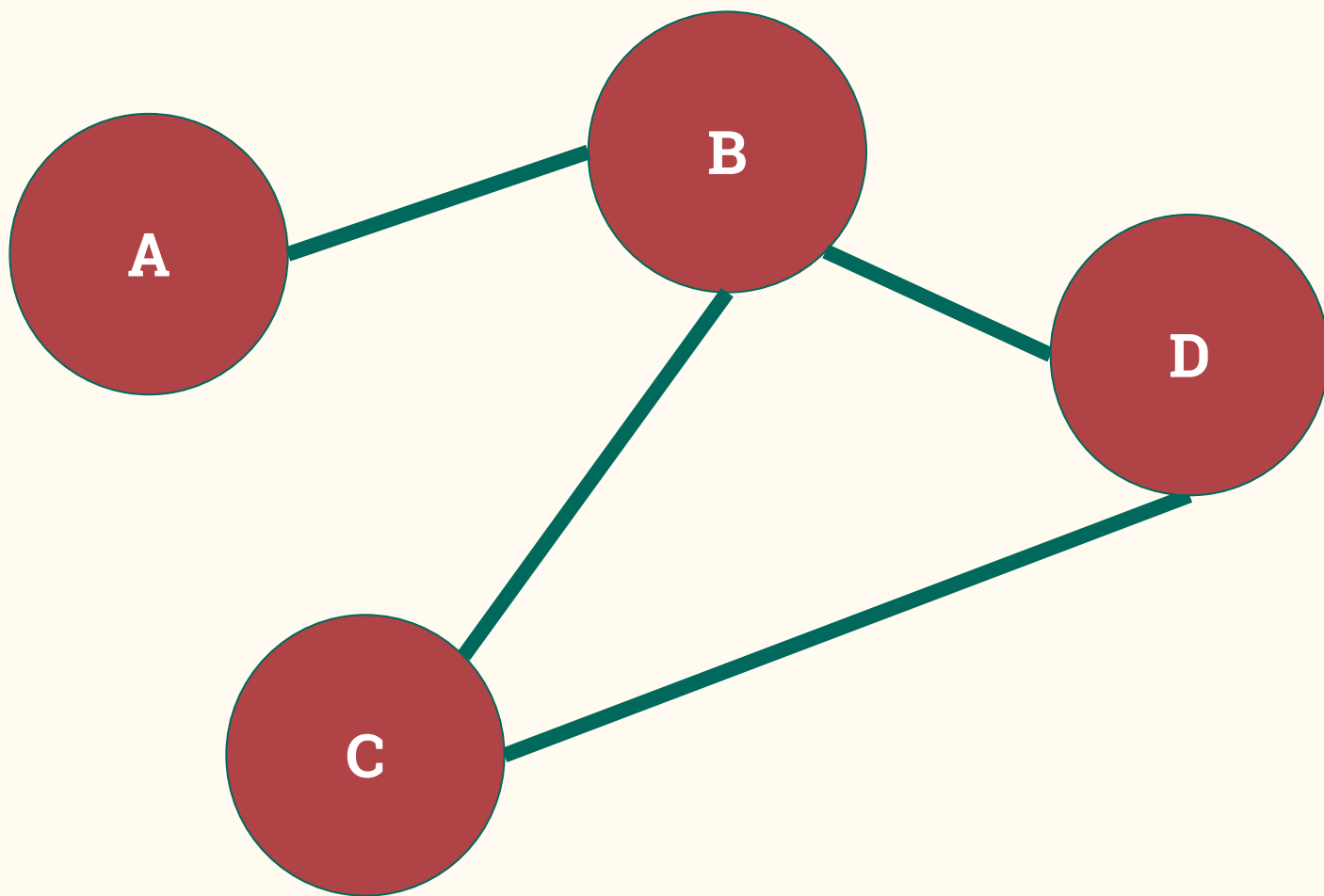
	Lista Ligada	Array	Lista Dinámica
Insert/delete at beginning	$\Theta(1)$	N/A	$\Theta(n)$
Insert/delete at end	$\Theta(n)$ when last element is unknown; $\Theta(1)$ when last element is known	N/A	$\Theta(1)$
Insert/delete in middle	search time + $\Theta(1)$	N/A	$\Theta(n)$

Grafos

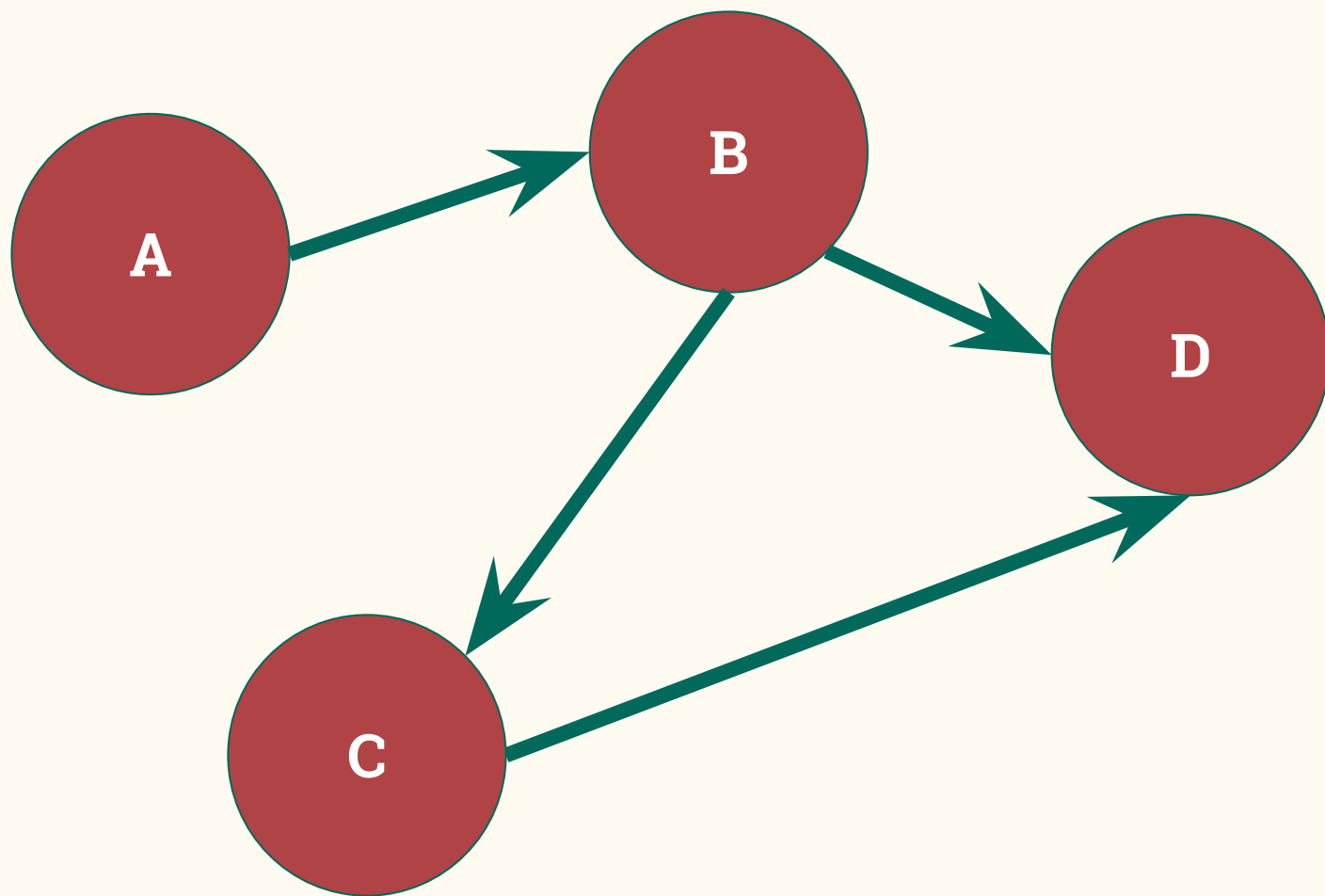


Estructuras de
datos que
representan
relaciones de
pares de objetos

No Dirigidos



Dirigidos



$G(V,E)$

- V , vértices y E aristas (edges)
- El **orden** es $|V|$
- El **tamaño** es $|E|$
- El **grado o valencia** de un vértice es el número de aristas conectadas a él

Representación

Lista

incidencia: cada arista es representada como un par ordenado de vértices

adyacencia: cada vértice tiene una lista de los que son incidentes a él

Matriz

incidencia: matriz $E \times V$ donde $[e, v]$ es uno si la arista con el vértice están conectados

adyacencia: matriz de $V \times V$ donde $[v1, v2]$ es uno si existe una arista entre los vértices