



# Complejidad Algorítmica

**Unidad 1: Comportamiento asintótico, métodos de búsquedas y grafos**

**Módulo 2: Algoritmos de Fuerza Bruta y  
Backtracking**

# Complejidad Algorítmica

Semana 2 / Sesión 1

## MÓDULO 2: Algoritmos de Fuerza Bruta y Backtracking



### Contenido

1. Problemas computacionales
2. Espacio o universo de búsqueda
3. Técnicas de búsqueda
  - Fuerza bruta (FB)
  - Backtracking (BT)
4. Ventajas y desventajas del Backtracking



### Preguntas

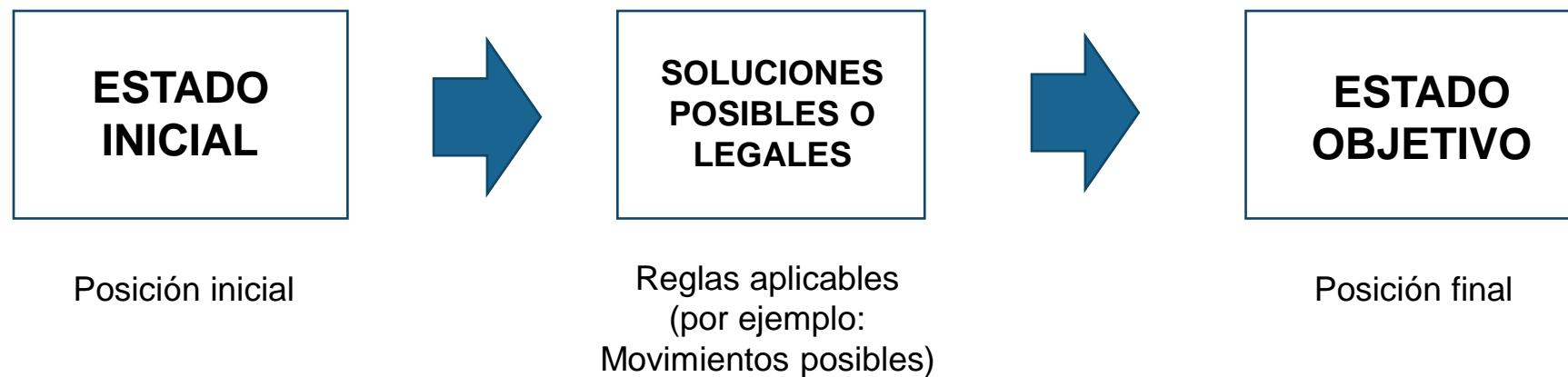
# 1. Problemas computacionales



- Los problemas son tareas que deben ser resueltas o cuyos objetivos de solución deben ser alcanzados.
- Existen problemas de todo nivel de complejidad: difíciles, medianamente difíciles y fáciles.
- Los informáticos resolvemos problemas a través del computador, por tanto, **debemos saber diseñar algoritmos que permitan encontrar una solución optima**.

# 1. Problemas computacionales

En computación, un problema lo definimos como un **espacio de estados (espacio de búsqueda)**, que incluye:



# 1. Problemas computacionales

## EJEMPLOS - PROBLEMAS COMPUTACIONALES

### 1.a. Problema 8-Puzzle

7	4	2
1		5
6	3	8

Estado Inicial

Reglas/Operadores/Acciones:

- Mover el recuadro vacío ARRIBA.
- Mover el recuadro vacío ABAJO.
- Mover el recuadro vacío IZQUIERDA.
- Mover el recuadro vacío DERECHA.

1	2	3
8		4
7	6	5

Estado Objetivo

# 1. Problemas computacionales

## EJEMPLOS - PROBLEMAS COMPUTACIONALES

### 1.b. Formulación del Problema 8-Puzzle

Estructura de datos: matriz 3x3

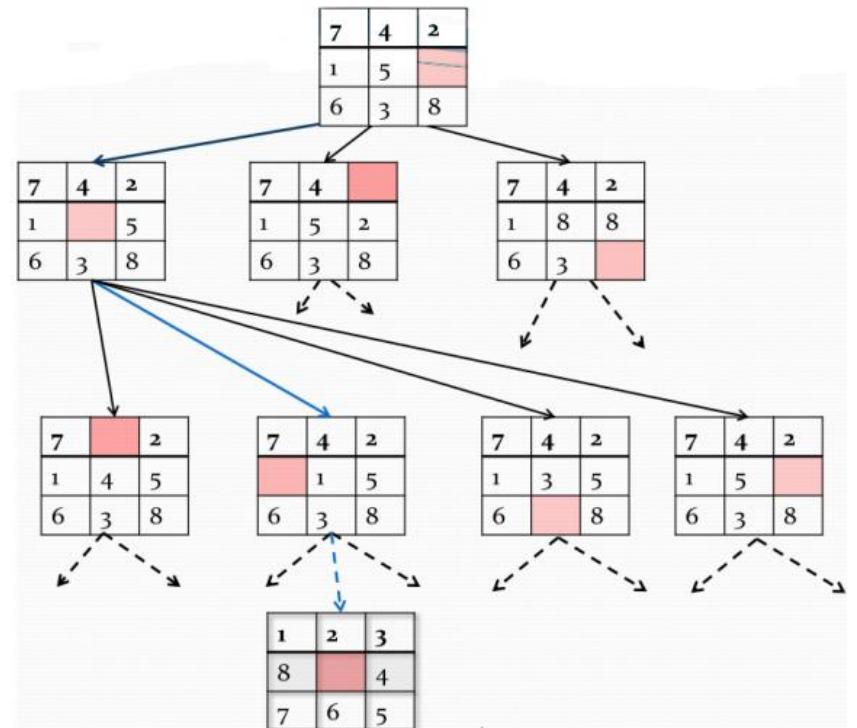
Recuadro vacío: representado con 'B'

Recuadros numerados: representados con su respectivo número.

7	4	2
1		5
6	3	8

representado por [[7, 4, 2][1, B, 5][6, 3, 8]]

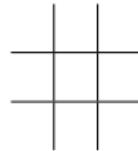
### 1.c. Espacio de solución de 8-Puzzle



# 1. Problemas computacionales

## EJEMPLOS - PROBLEMAS COMPUTACIONALES

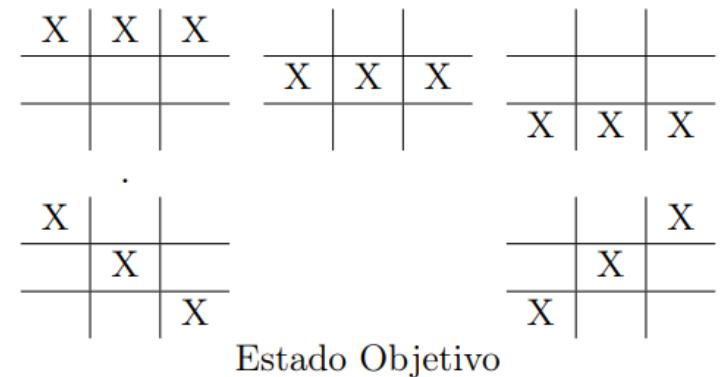
### 1.a. Problema: 3 en raya



Estado Inicial

Reglas/Operadores/Acciones:

- Jugador 1 marca X.
- Jugador 2 marca O.



# 1. Problemas computacionales

## EJEMPLOS - PROBLEMAS COMPUTACIONALES

### 1.b. Formulación del Problema 3 en raya

Estructura de datos: matriz 3x3

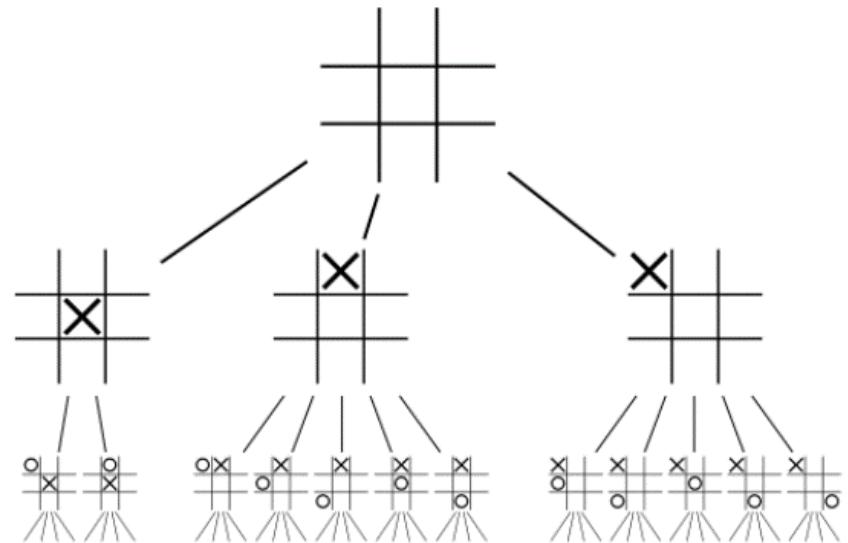
Marcado del jugador 1: representado con 'X'

Marcado del jugador 2: representados con 'O'.

		X
	X	
X		

representado por `[["","'X'"],["'X','"],['X','','']]`

### 1.c. Espacio de solución de 3 en raya



## 2. Espacio o universo de búsqueda

### ESPACIO DE BUSQUEDA (UNIVERSO DE BUSQUEDA o ESPACIO DE ESTADO)

¿Qué es un espacio de búsqueda?



- Son todas las posibilidades de respuesta ante un problema, sean estas correctas o no.
- Permite encontrar la respuesta correcta a un problema (según la técnica que apliquemos).
- Los problemas de optimización también pueden ser considerados como problemas con espacio de búsqueda.

## 2. Espacio o universo de búsqueda

### RESOLUCION DE PROBLEMAS COMPUTACIONALES

Un problema puede ser resuelto a través de:

- El análisis.
- La definición de sus características esenciales.
- La aplicación de una técnica conocida para ese problema.

Las (2) técnicas clásicas de resolución de problemas o técnicas de búsquedas son:

1. **Fuerza bruta** (búsqueda exhaustiva)
2. **Backtracking**



### 3. Técnicas de búsquedas

¿Qué es Fuerza Bruta?



- Es una estrategia sencilla pero a la que le hace falta sofisticación en la solución.
- Busca sistemáticamente en todo el espacio de búsqueda.
- Se toma la ruta más directa, sin ningún intento de minimizar el número de operaciones necesarias para calcular la solución (a lo bruto?)
- La complicación viene en poder generar todas las soluciones candidatas.
- Esta técnica, generalmente traduce directamente el enunciado del problema.
- A menudo es “suficientemente buena” especialmente para pequeñas instancias.

### 3. Técnicas de búsquedas

#### Fuerza Bruta

- Una estrategia general para listar todas las opciones en un problema de Fuerza Bruta es usar **permutaciones o combinaciones**.

**Combinaciones** con o sin repetición → Si el orden no importa, es una **combinación**.

**Permutaciones** con o sin repetición → Si el orden **sí** importa es una **permutación**.



Una permutación es una combinación ordenada.

### 3. Técnicas de búsquedas

#### Aplicaciones de la técnica de búsqueda de Fuerza Bruta

Son ejemplos de Fuerza Bruta:

- La búsqueda de clasificación por selección (Selection Sort)
- La búsqueda de ordenamiento por burbuja (Bubble Sort)
- Coincidencia de cadenas (String Matching o la ocurrencia de una cadena en un texto)

---

#### HOW A BASIC BRUTE FORCE ATTACK WORKS

---



Attacker



Guess List Of  
Username &  
Password  
Combination



Repeats Login  
Attempts Until  
One Is Successful



Successful  
Credential Validation

### 3. Técnicas de búsquedas

#### TECNICA DE FUERZA BRUTA - EJEMPLOS

##### Ejemplo #1

Buscar un producto específico en el supermercado en donde no conocemos la distribución de sus productos, solo los tipos de productos.



##### Solución:

###### a) Sofisticada:

Si el producto es un alimento congelado, de inmediato se va al pasillo donde se encuentran los alimentos congelados, sin recorrer otro pasillo.

###### b) Fuerza Bruta:

Fuerza Bruta ignora la alternativa anterior e ingenuamente se busca a través de todas los pasillos en el intento de encontrar el producto deseado.

### 3. Técnicas de búsquedas

#### TECNICA DE FUERZA BRUTA - EJEMPLOS

##### Ejemplo #2

Encontrar cierto elemento en una vector de tamaño  $n$ , asumir que el vector no está ordenado.



##### Solución:

```
función lineal (A, n, key : entero ) es
    desde i :=1 hasta n hacer
        si A[i] = key entonces
            retornar i ;
        fin si
    fin desde
    retornar -1;
fin función
```

### 3. Técnicas de búsquedas

#### TECNICA DE FUERZA BRUTA - EJEMPLOS

##### Ejemplo #3

Implementar la **ordenación por selección**:

- Buscar el mínimo elemento de la lista
- Intercambiarlo con el primero
- Buscar el siguiente mínimo en el resto de la lista
- Intercambiarlo con el segundo



##### Solución:

50	26	7	9	15	27
7	26	50	9	15	27
7	9	50	26	15	27
7	9	15	26	50	27
7	9	15	26	27	50

Si el orden **sí** importa es una **permutación**.

### 3. Técnicas de búsquedas



- **BACKTRACKING** es una técnica para **buscar** sistemáticamente a través de todas las configuraciones de un espacio de búsqueda.
  - Para esto se construye todas las posibles soluciones candidatas de manera sistemática.
  - Dada una solución candidata **S**, se procede a
    1. Verificar si **S** es solución. Si lo es, hacen algo con ella (depende del problema).
    2. Construyen todas las posibles extensiones de **S**, e invocan recursivamente al algoritmo con todas ellas.

### 3. Técnicas de búsquedas

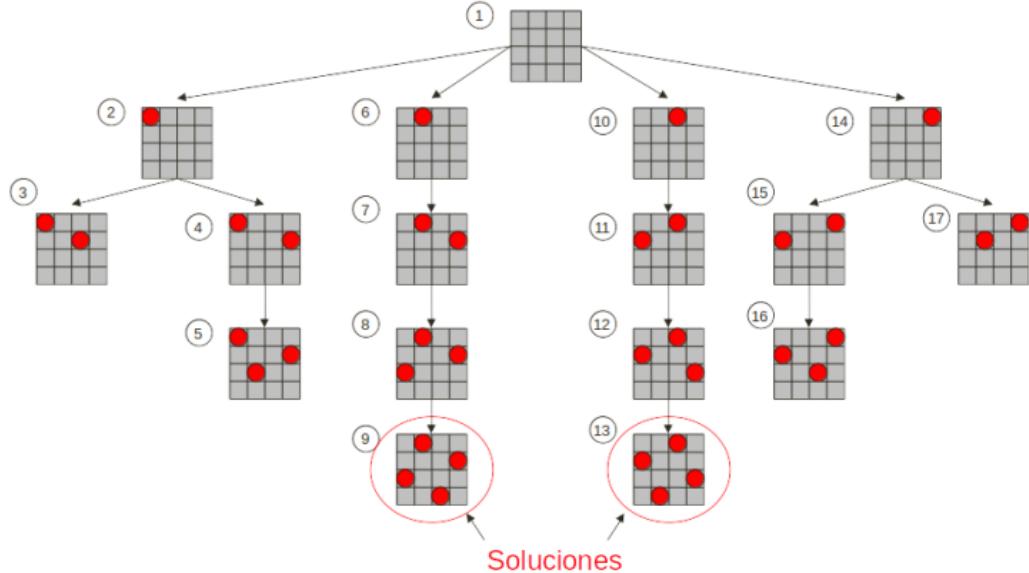
#### BACKTRACKING: Procedimiento general

```
PROCEDURE VueltaAtras(etapa);
BEGIN
    IniciarOpciones;
    REPEAT
        SeleccionarNuevaOpcion;
        IF Aceptable THEN
            AnotarOpcion;
            IF SolucionIncompleta THEN
                VueltaAtras(etapa_siguiente);
                IF NOT exito THEN
                    CancelarAnotacion
                END
            ELSE (* solucion completa *)
                exito:=TRUE
            END
        END
    UNTIL (exito) OR (UltimaOpcion)
END VueltaAtras;
```

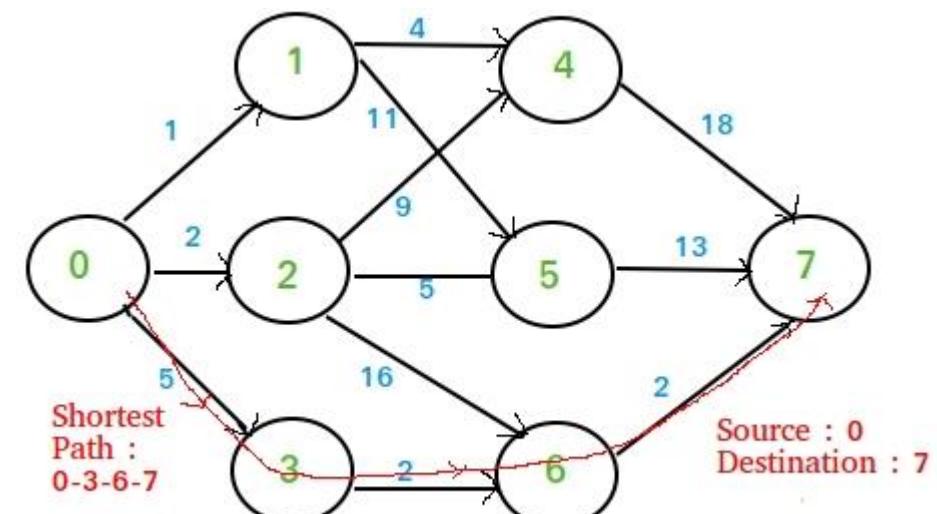
### 3. Técnicas de búsquedas

#### EJEMPLOS DE LA TECNICA BACKTRACKING

- 4 REINAS



- PROBLEMA DEL VIAJERO



## 4. Ventajas y desventajas del Backtracking

### VENTAJAS

- El Backtracking puede resolver casi cualquier problema, debido a su naturaleza de fuerza bruta.
- Se puede utilizar para encontrar todas las soluciones existentes si existe algún problema.
- Es una representación paso a paso de una solución a un problema dado, que es muy fácil de entender.
- Muy fácil de escribir el código y también de depurar.

### DESVENTAJAS

- Es muy lento en comparación con otras soluciones.
- Dependiendo de los datos que tenga, existe la posibilidad de realizar una búsqueda muy grande con Backtracking y al final no encontrar ninguna coincidencia con sus parámetros de búsqueda.
- Costo computacional muy alto, Backtracking es un algoritmo recursivo que consume mucho de la memoria y del procesador.
- Considerar usar el algoritmo de ramificación y poda (para gran volumen de datos, consume menos recursos a un coste de tiempo menor).

# CONCLUSIONES

- El espacio de búsqueda es el conjunto de todos los estados posibles (SOLUCIONES) para un problema dado.
- Técnicas de búsqueda son las formas de tratar el espacio de búsqueda.
  - Fuerza bruta
  - Backtracking
- La **Complejidad Algorítmica** variará de acuerdo a la técnica utilizada y el tamaño del espacio de búsqueda (cantidad de posibles soluciones).

# PREGUNTAS

Dudas y opiniones