

# Web Assembly

## IIC 3585-1

Grupo 4 – Eich



# WEBASSEMBLY

- Vicente Chadwick, Felipe Trejo, Lucas Rodríguez

# Problema del vendedor viajero

- Múltiples nodos conectados entre sí
- Se puede llegar de cualquier nodo a todos los otros
- Recorrer todos los nodos con la menor suma de "pesos" posible



Algoritmo usado



# Programación Dinámica

$O(2^n \times n^2)$

vs

## "Fuerza Bruta"

$O(n!)$

# Programación Dinámica

En palabras simples

Sub-Problema 1

- $A \rightarrow B \rightarrow \{C, D, E\}$

**Calculamos**

$$\{C, D, E\} = 20$$

Sub-problema 2

- $B \rightarrow A \rightarrow \{C, D, E\}$



Tomemos  $n=7$



# Programación Dinámica

= 177

vs

## "Fuerza Bruta"

= 5040

Tomemos  $n=10$



# Programación Dinámica

= 102.400

vs

## "Fuerza Bruta"

= 3.628.800

# WASM

```
$( "#create-path" ).click(function() {  
  Module.onRuntimeInitialized = async _ => {  
    const api = {  
      version: Module.cwrap('main_fun', 'number', ['number', 'string']),  
    };  
    const matrix_to_string = create_matrix();  
    api.version(data.nodes.length, matrix_to_string.trim())  
  };  
  Module.onRuntimeInitialized();  
});
```

# Función principal C



```
uint32_t main_fun(uint32_t n, char distS[])
{
    int m = pow(2, n);

    uint32_t dp[m][n][n + 1];

    // Se crea array y con funcion convert se le asignan valores
    uint32_t dist[n][n];
    convert(distS, n, dist);

    // Inicializar Base de Datos Dinamica
    for (uint32_t i = 0; i < (1 << n); i++)
    {
        for (uint32_t j = 0; j < n; j++)
        {
            dp[i][j][0] = -1;
        }
    }

    // OUTPUT
    // *****
    // Esto sera la solucion del programa
    uint32_t route[n];
    uint32_t cost = INT_MAX;
    // *****
```



# Función principal C

```
// Resolvemos el problema partiendo desde diferentes nodos
for (uint32_t start = 0; start < n; start++)
{
    // Inicializamos un array que nos dara el return de la funcion, arr[0] = costo total, arr[1..n] = ruta
    uint32_t arr[n + 1];
    for (uint32_t i = 0; i < n + 1; i++)
        arr[i] = -1;

    // Resolvemos el problema partiendo desde un cierto nodo
    tsp(1 << start, start, arr, n, dist, dp);

    // Evaluamos si es la mejor solucion
    if (arr[0] < cost)
    {
        cost = arr[0];
        for (uint32_t i = 0; i < n; i++)
            route[i] = arr[i + 1];
    }
}
```

Ahora veamos el programa...

# Web Assembly

## IIC 3585-1

Grupo 4 – Eich



# WEBASSEMBLY

- Vicente Chadwick, Felipe Trejo, Lucas Rodríguez