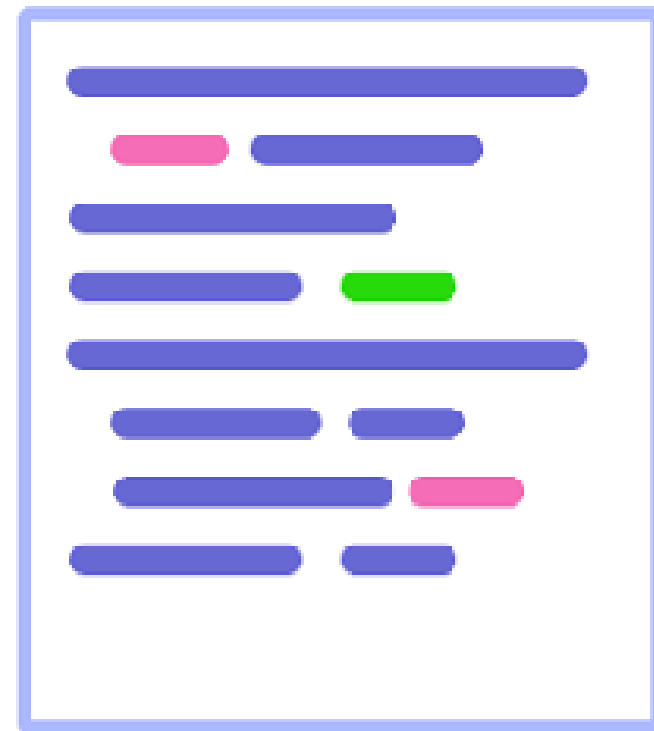
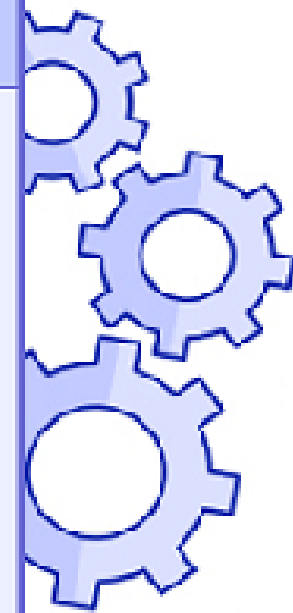
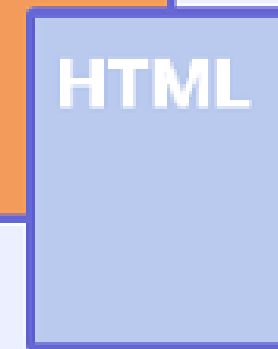
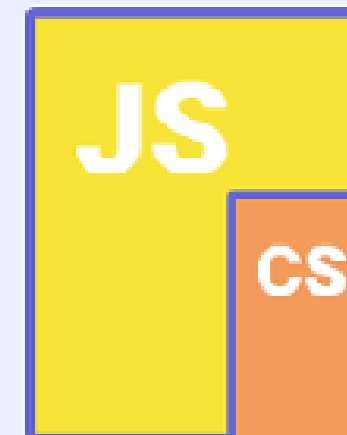


C++, C or Rust



+



Benjamín Ayancán
Nicolás Benitez
Fernanda Durán

Web Assembly

Índice

1. Tarea

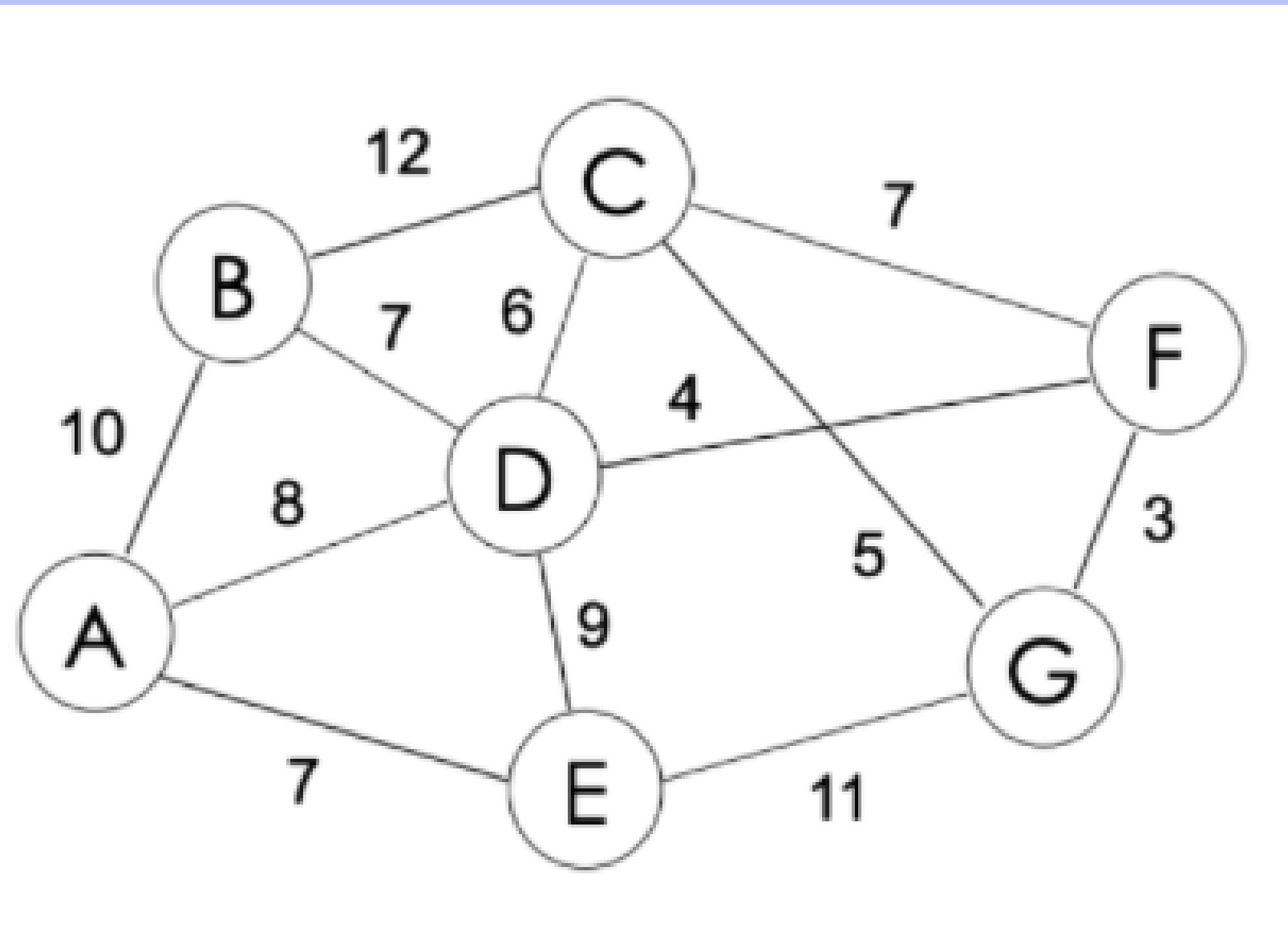
2. Demo

3. Código

4. Dificultades

5. Aprendizajes

Tarea



Implementar un programa que encontrara la mejor ruta para un vendedor dado que tiene que visitar a todos sus clientes.

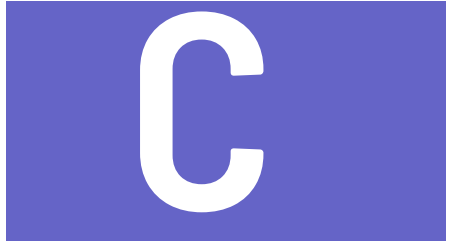
La figura muestra la ubicación de los clientes y las distancias asociadas a cada camino disponible.

El problema consiste en encontrar la secuencia óptima.

.

Demo

Código



Función least

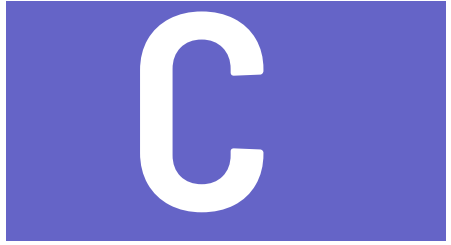
```
int ary[MAX][MAX],completed[MAX],n,cost;

int least(int c, int** ary)
{
    int i,nc=999;
    int min=999,kmin;

    for(i=0;i < n;i++)
    {
        if((ary[c][i]!=0)&&(completed[i]==0))
            if(ary[c][i]+ary[i][c] < min)
            {
                min=ary[i][0]+ary[c][i];
                kmin=ary[c][i];
                nc=i;
            }
    }

    if(min!=999)
        cost+=kmin;

    return nc;
}
```



Función mincost

```
void mincost(int city, int* memoria, int k, int** ary)
{
    int i, ncity;

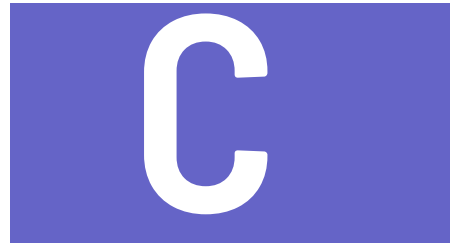
    completed[city]=1;

    printf("%d--->", city+1);
    memoria[k] = city+1;
    ncity = least(city, ary);

    if(ncity==999)
    {
        ncity=0;
        printf("%d", ncity+1);
        cost+=ary[city][ncity];

        return;
    }

    mincost(ncity, memoria, k+1, ary);
    memoria[0] = cost;
}
```



Función tsp

```
int tsp(int* memoria, int** ary, int dimension)
{
    cost = 0;
    n = dimension;
    int i;
    for(i=0;i < n;i++)
    {
        completed[i]=0;
    }

    printf("\n\nThe Path is:\n");
    mincost(0, memoria, 1,ary); //passing 0 because starting vertex

    printf("\n\nMinimum cost is %d\n ",cost);

    return 0;
}
```


Javascript

Función makePtrOfArray

```
const makePtrOfArray = (myModule, N, graph) => {  
  const arrayPtr = myModule._calloc(N, 4);  
  for (let i = 0; i < N; i++) {  
    let rowsPtr = myModule._calloc(N, 4);  
    myModule.setValue(arrayPtr + i * 4, rowsPtr, "i32");  
    for (let j = 0; j < N; j++) {  
      myModule.setValue(rowsPtr + j * 4, graph[i][j], "i32");  
    }  
  }  
  return arrayPtr;  
};
```

Javascript

Función getArrayFromPtr

```
const getArrayFromPtr = (myModule, ptr, N) => {  
  let resultMatrix = [];  
  for (let i = 0; i < N; i++) {  
    resultMatrix[i] = myModule.getValue(ptr + i * 4, "i32");  
  }  
  return resultMatrix;  
};
```

Javascript

Función cRunner

```
const cRunner = (Module, arrayPtr, G, dim) => {  
  let startTime = window.performance.now();  
  let result = Module._tsp(arrayPtr, G, dim);  
  let endTime = window.performance.now();  
  console.log("result", result);  
  const resultTime = endTime - startTime;  
  return resultTime;  
};
```

Javascript

```
const calcBtn = document.getElementById("calc-btn");
const matrixInput = document.getElementById("matrix-input");

Module().then(function (mymod) {
  calcBtn.onclick = () => {
    const parsedInputGraph = JSON.parse(matrixInput.value);
    const dim = parsedInputGraph[0].length;
    const dimResult = dim + 1;
    const arrayPtr = mymod._calloc(100 * dim, 4);
    const G = makePtrOfArray(mymod, dim, parsedInputGraph);
    let cTime = cRunner(mymod, arrayPtr, G, dim);
    let matrix = getArrayFromPtr(mymod, arrayPtr, dimResult);
    console.log(`cresult - ctime ${matrix} ${cTime}`);
    addToTable(matrix, cTime);
  };
});
```

Javascript

Función addToTable

```
const addToTable = (resultArr, execTime) => {  
  const table = document.getElementById("result-table");  
  let row = table.insertRow(-1);  
  let cell1 = row.insertCell(0);  
  let cell2 = row.insertCell(1);  
  let cell3 = row.insertCell(2);  
  cell1.innerHTML = `${resultArr[0]}`;   
  cell2.innerHTML = `${resultArr.slice(1).join("->")}`;   
  cell3.innerHTML = `${execTime} ms`;   
};
```

Dificultades

- Tuvimos muchos errores al momento de programar por lo que nos fuimos quedando atascados error tras error.
- No lograbamos recibir el output nuestro código de C, siendo el uso de memoria uno de los pasos más complejos de manejar.

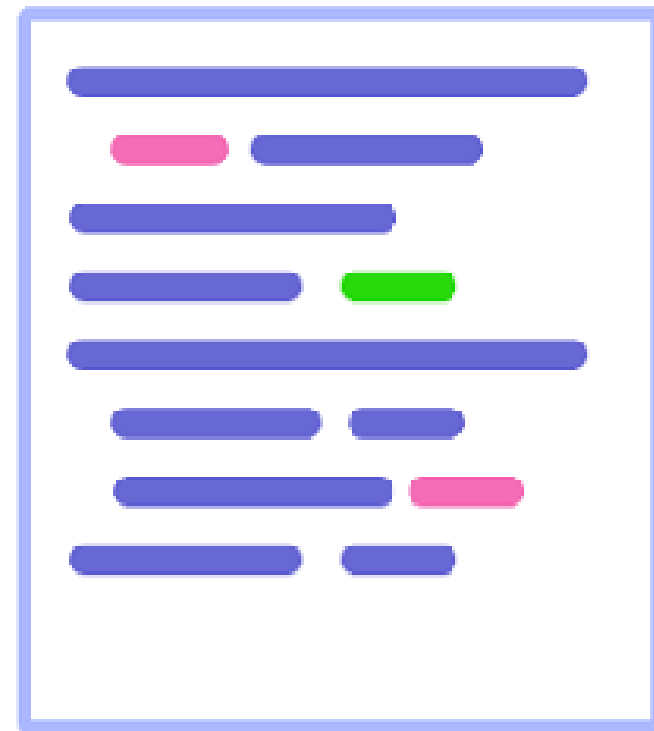


Aprendizajes

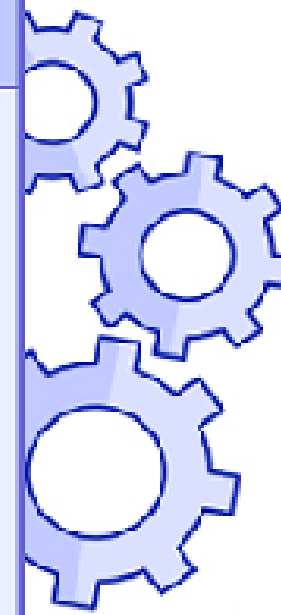
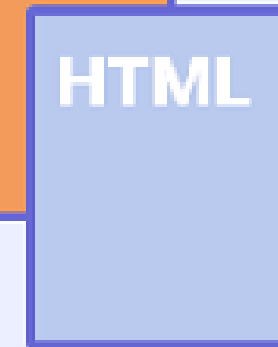
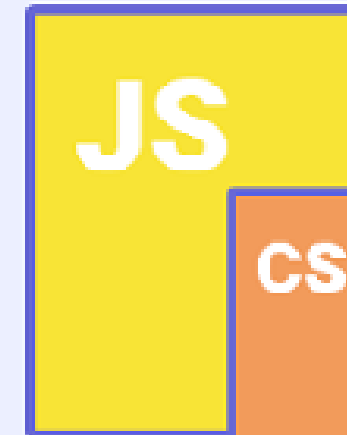
- Logramos comprender la base de Web Assembly, programar una conexión entre ellos y poder aprovechar las ventajas de C



C++, C or Rust



+



Benjamín Ayancán
Nicolás Benitez
Fernanda Durán

Muchas gracias