WASM

Grupo 2

BESOAIN,
CARRASCO,
JIMENEZ

ALGORITMO A UTILIZAR

- **Setear clusters en tiempo = 0**
- **Iniciar iteración**
- **Buscar el tiempo minimo entre los clusters candidatos**
- **Setear en el cluster**
- **Repetir**

```
    for (let i = 0; i < numClusters; i++) {
        clustersTimes.set(i, 0);
    }

    for (let i = 0; i < times.length; i++) {
        const time = times[i];
        let minTime = Infinity;
        let clusterId;

        for (const [id, totalTiempo] of clustersTimes.entries()) {
            if (totalTiempo < minTime) {
                minTime = totalTiempo;
                clusterId = id;
            }
        }

        clusters[clusterId][i] = i+1;
        clustersTimes.set(clusterId, minTime + time);
    }
```

PASOS A SEGUIR

1

Incoveniente al usar librerias como vector, multiset.

2

Actualizar codigo a C,

3

Realizar interfaz para presentación

PROCESO DE TRABAJO

Cwrap vs Ccall

Cwrap

Toma una función y le hace wrap devolviendo una función de javascript
Mejor para llamar la función repetidas veces

Ccall

Llama a la funcion compilada en C y retorna su valor

Codigo para compilar a emscripten

```
emcc -sEXPORTED_FUNCTIONS=_malloc,_free,_calloc -sEXPORTED_RUNTIME_METHODS=cwrap
 -O3 -o function.js function.c
```

OnRuntimeInitialized

Funcion que permite que la función sea llamada cuando el runtime este totalmente inicializado.

Nos solucionó problemas de sincronía donde no recibiamos valores en consola.

Llamado a funciones

```
const timePtr = Module._malloc(timesArray.byteLength);
Module.HEAPU32.set(timesArray, timePtr >> 2);
const optimizeFunc = Module.cwrap('optimize', 'number', ['number', 'number', 'number']);
```

Calculo de tiempo

```
const startTimeC = performance.now();
const clustersPtr = optimizeFunc(timePtr, times.length, clusterNum);
const endTimeC = performance.now();
const timeC = endTimeC - startTimeC;
```

Obtener valores

```
for (let i = 0; i < clusterNum; i++) {
  const clusterPtr = Module.getValue(clustersPtr + i * 4, 'i32');
  const cluster = [];

  for (let j = 0; j < times.length; j++) {
    const value = Module.getValue(clusterPtr + j * 4, 'i32');
    if (value === -1) break;
    cluster.push(value);
  }

  clusters.push(cluster);
}
```

DEMO

CONCLUSIONES Y SUPOSICIONES

Tiempo de C en la aplicación
Utilidad de WASM y funcionalidades