Universidad Nacional de Colombia
Facultad de Ingeniería
Computación paralela y distribuida
Docente: Oscar Agudelo Rojas
Estudiante: Anderson Stick Barrera Tovar

Actividad:

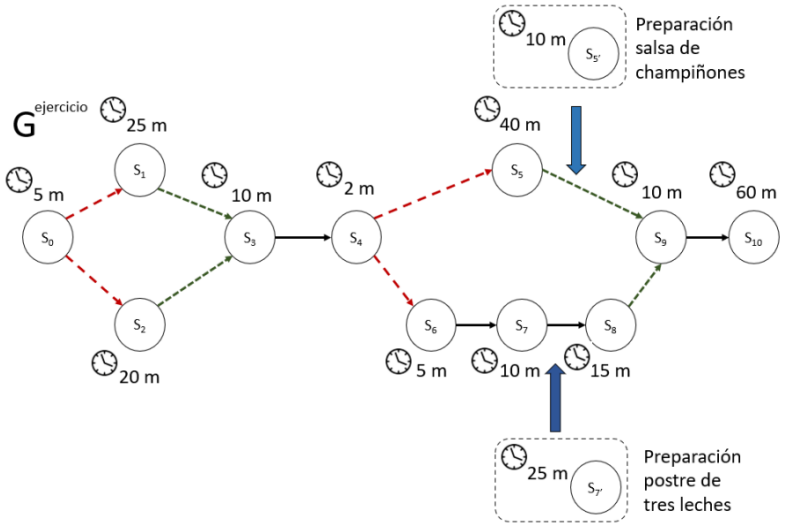
a) Averiguar ¿Qué es un POSET (Patially Ordered Set)?

Un POSET, abreviatura de Conjunto Parcialmente Ordenado, se refiere a un conjunto no vacío en el cual se ha establecido una relación de orden parcial. Esta relación binaria cumple con ciertas características:

- **Reflexividad:** Cada elemento del conjunto está relacionado consigo mismo.
- **Antisimetría:** Si dos elementos están relacionados mutuamente, entonces son iguales.
- **Transitividad:** Si un elemento está relacionado con otro y ese segundo elemento está relacionado con un tercero, entonces el primero también está relacionado con el tercero.

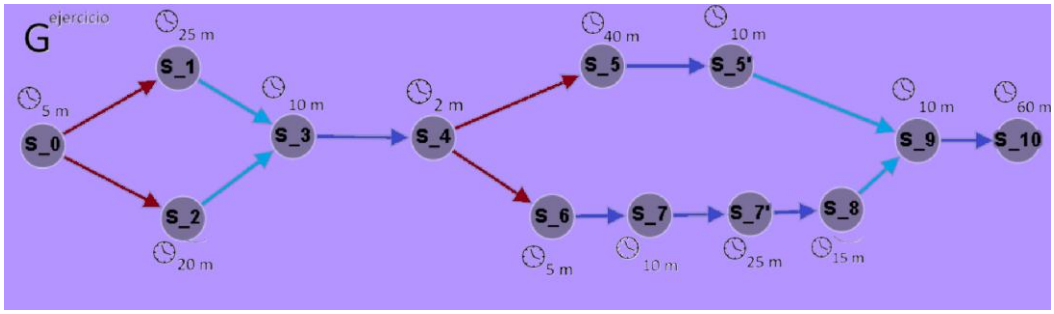
En esencia, un POSET es un conjunto en el cual algunos elementos pueden considerarse "mayores" que otros, pero no todos los elementos pueden ser comparados entre sí. Por ejemplo, el conjunto de números naturales con la relación "divisible por" es un POSET, ya que algunos números son divisibles entre sí, pero no todos los números son comparables (por ejemplo, no se puede comparar 3 y 5).

b) Con base en la siguiente figura:



i. Al grafo computacional $\boldsymbol{G^{Ejercicio}}$ adiciónele las dos actividades indicadas en el sitio que señalan las flechas azules en el diagrama. Después calcule el $\boldsymbol{WORK}(\boldsymbol{G^{Ejercicio}})$, el $\boldsymbol{SPAN}(\boldsymbol{G^{Ejercicio}})$ y el paralelismo ideal correspondiente.

Nuevo grafo computacional (Añadiendo S5' y S7')



Ahora se encuentra el $\boldsymbol{WORK}(\boldsymbol{G^{Ejercicio}})$ el cual es la suma de los tiempos de ejecución de cada tarea

$$WORK(G^{Ejercicio}) = 5m + 25m + 20m + 10m + 2m + 40m + 10m + 5m + 10m + 25m + 15m + 10m + 60m = 237\ m$$

Cálculo del $\boldsymbol{SPAN}(\boldsymbol{G^{Ejercicio}})$

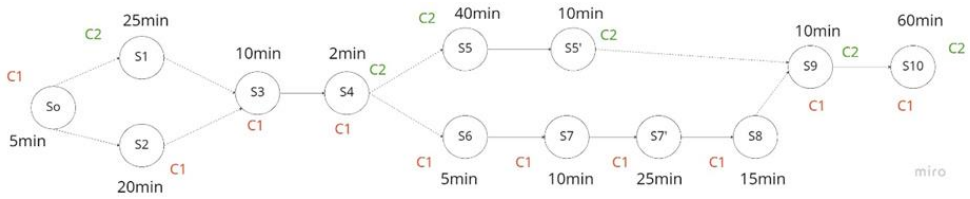
$$SPAN\ (G^{Ejercicio}) = 5m + 25m + 10m + 2m + 5m + 10m + 25m + 15m + 10m + 60m = 167\ m$$

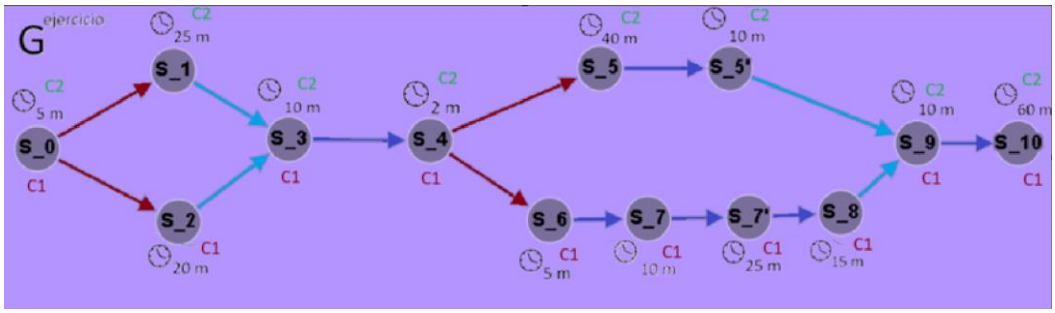
Finalmente, cálculo del paralelismo ideal

$$SPEEDUP = \frac{WORK(G^{Ejercicio})}{SPAN(G^{Ejercicio})} = \frac{237}{167} = 1.41916$$

ii. Escribir el pseudocódigo de inicio y terminación de tareas con async y finish.

Se realiza la distribución de las tareas en dos CORES, quedando la siguiente gráfica





El pseudocódigo de inició a terminación quedaría:

```
1   {
2       S_0
3       finish {
4           S_2
5           async {
6               S_1
7           }
8       }
9       S_3
10      S_4
11      finish {
12          S_6
13          S_7
14          S_7'
15          S_8
16          async {
17              S_5
18              S_5'
19          }
20      }
21      S_9
22      S_10
23  }
```

c) El siguiente seudo-código suma dos matrices triangulares inferiores (matrices cuadradas n x n en las cuales los elementos por encima de la diagonal, incluyendo la diagonal de (0,0) a (n,n), son cero). En el código, cada ejecución de la sentencia A[i][j] = B[i][j] + C[i][j]; representa una unidad de tiempo en el grafo computacional correspondiente.

```
finish {
        for (int i = 0; i < n;  i++) {
                async {
                        for (int j = 0; j < i; j++) {
```

```
                    A[i][j] = B[i][j] + C[i][j];
                } // for -j
            } // async
        } // for-i
    } // finish
```

- **En términos de n ¿Cuál es el WORK del código mostrado? ES necesario que explique cómo obtuvo su respuesta**

  Para calcular el $WORK$ del algoritmo anteriormente mencionado, se debe tener en cuenta que el $WORK$ se refiere al tiempo total de ejecución del programa en términos de la cantidad de operaciones que realiza cada Core, se observa en el código se observa que hay dos Cores, uno que realiza el ciclo (For) de **1** a **n**, teniendo **n** operaciones y el segundo que ejecuta una **sumatoria de n** operaciones, por lo que podemos expresar la cantidad de operaciones del Core 1 y del Core 2 como

$$Core\ 1 = n$$

$$Core\ 2 = \sum_{i=1}^{n} i$$

  Por lo tanto, teniendo en cuenta que $WORK = Core\ 1 + Core\ 2$ tenemos que:

$$WORK = n + \sum_{i=1}^{n} i$$

REFERENCIAS:

1. McCool, M., Reinders, J., & Robison, A. (2012b). Structured Parallel Programming: Patterns for Efficient Computation. Elsevier Gezondheidszorg. Recuperado 1 de marzo de 2023, de http://arcesio.net/paralela/libros/Structured_Parallel_Programming_Patterns_for_Efficient_Computation_2012.pdf

2. Reggiani, S. (2018, septiembre). Conjuntos parcialmente ordenados. fceia. Recuperado 1 de marzo de 2023, de https://www.fceia.unr.edu.ar/~reggiani/files/ag2/2018-09-18-Posets.pdf