PARTE 3: Ejercicio de programación JAVA

Duración 70 minutos

Este ejercicio comprende **2** pasos:

1. Desarrollo de la funcionalidad especificada más abajo
2. Desarrollo de los casos de prueba ("test case", al menos uno) para verificar la corrección de la funcionalidad implementada.

ESCENARIO

Es común, en múltiples procesos de computación, que éstos puedan ser definidos y descompuestos en un cierto conjunto de tareas – o **"tasks"** – que, en el momento de la ejecución, podrían eventualmente paralelizarse o ejecutarse en forma concurrente para lograr mayor eficiencia. Estas tareas habitualmente tendrán relaciones de dependencia fuertes, es decir, algunas tareas necesitarán que otra(s) hayan culminado para poder ejecutarse.

Para un cierto proceso de cómputo, conocemos la información de cada una de las tareas que lo componen: Nombre (ej: "Task1") y tiempo de ejecución (ej.: 1 – podemos asumir un valor entero mayor que 0, dado en milisegundos-).

También tenemos una lista de "**requerimientosPrevios**", en que se indica, para cada tarea, todas aquéllas otras que es necesario ejecutar previamente para poder iniciar la tarea.

Para simplificar, podríamos asumir que el proceso comienza en un estado "**Start**", (con **duración 0**, único que no requiere de otra tarea previa , y termina en un estado "**End**" (también **duración 0**, y al cual no le sucede ninguna otra tarea).
Deseamos saber también cuál será la "**secuencia crítica de ejecución del proceso**", es decir, aquélla, **entre todas las secuencias posibles (desde "Start" hasta "End")** en la cual cualquier retraso (en cualquiera de los "tareas" que la componen – por ejemplo ocasionado por demora en acceso a los datos -) determinará que completar el proceso requiera ***indefectiblemente*** más tiempo (ver diagrama ejemplo al reverso).

PASO 1: Funcionalidad a desarrollar:

Descargar el proyecto NETBEANS "Parcial2.zip" que contiene clases base de Grafos requeridas para este trabajo (que pueden ser extendidas y modificadas). Observar la clase ***TProceso*** que modela el problema.
La clase "SecuenciaCriticaTareas" tiene el método "main" para ejecutar el programa, completar.
Utilizar el TProceso (extiende TDA Grafo Dirigido) para representar este problema. Las "**tasks**" (archivo "tasks.txt") serán los "**vértices**" del grafo, y tendrán una etiqueta ("Nombre de la tarea"), y una "**duración**"). La etiqueta es un comparable –string- y la duración un entero positivo. Existirá un vértice especial "**Start**" y uno "**End**", ambos con duración 0.

Las relaciones de requerimientos de ejecución previa serán representadas mediante los arcos del grafo (archivo "links.txt"). En este caso, el costo del arco no tiene importancia ni utilidad.

Se requiere entonces:

1. Generar una estructura apropiada para representar el problema y cargarla a partir de los archivos de entrada indicados. Ver detalles de los archivos de entrada al reverso.
2. Desarrollar una funcionalidad que, dado un grafo que represente las tareas y sus relaciones de dependencias, devuelva – y muestre por consola - la secuencia de tareas crítica de un proceso (listado de las tareas componentes, cada cual con su duración) y el tiempo total mínimo que insumiría el proceso.
3. Implementar las verificaciones básicas que aseguren que la funcionalidad anterior se puede ejecutar correctamente para el grafo representado.
4. Ejecutar el programa con el juego de datos provisto
5. Indicar qué precondiciones (o precauciones a tomar) deben ser implementadas.

PASO 2: TEST CASE.

Implementa el o los **Casos de Prueba** necesarios para verificar el correcto funcionamiento del método desarrollado.

ENTREGA

Subir a la webasignatura, en la tarea "**PARCIAL2-PARTE3**" un archivo "parcial.zip" que contenga todo el proyecto más un archivo **"salida.txt"** que tenga la salida de consola de la ejecución.

ARCHIVOS DE ENTRADA

"TASKS.TXT"
Una "task" por línea, con la siguiente estructura:

- **Nombre de task (string) , duración (entero)**

Ejemplo:

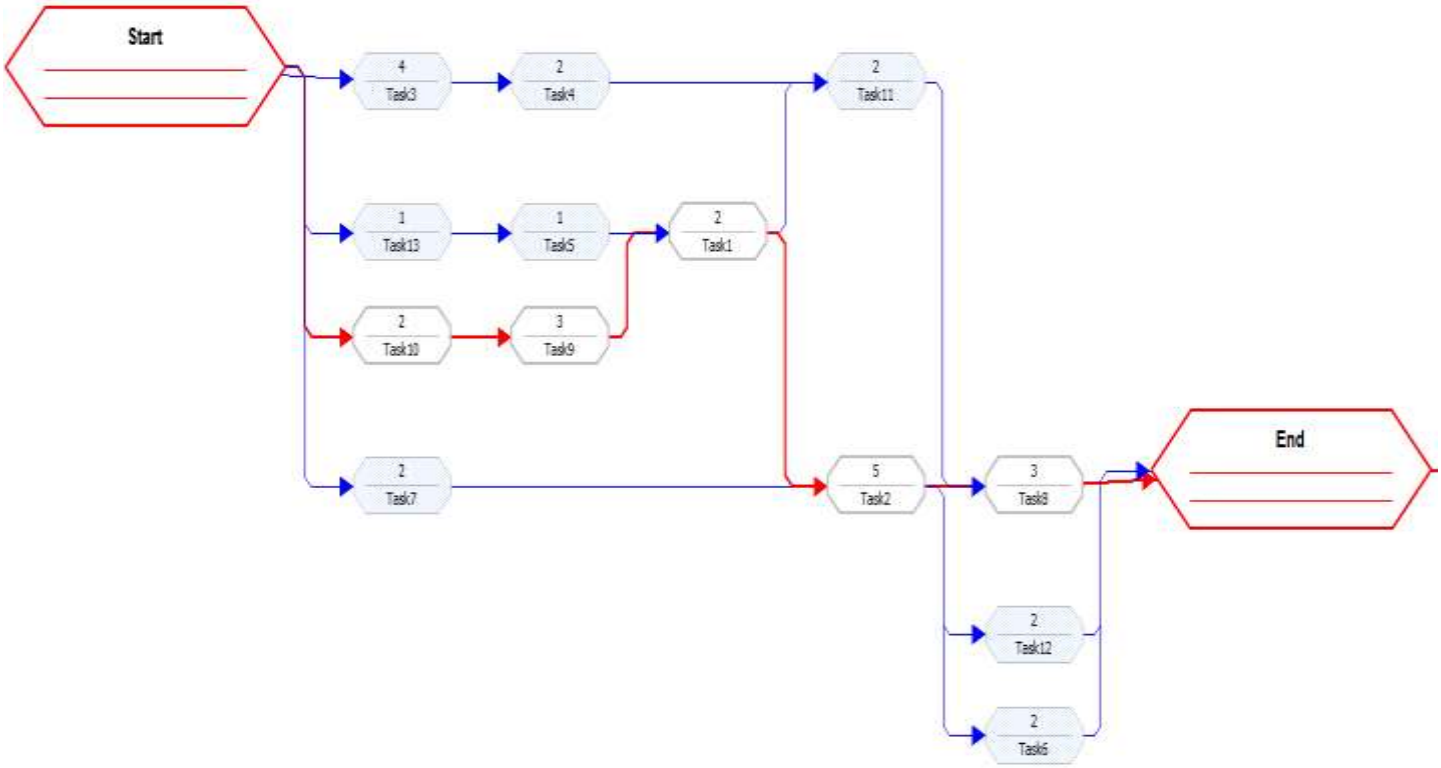
Task1 , 1
Task3, 4
Start, 0
End, 0

"LINKS.TXT"

- **"Task" previa** (string)**, "Task" actual** (string), 0

Ejemplo:

Task13, Task5,0
Start, Task13,0
Task6, End,0



En este ejemplo la secuencia crítica es Start-Task10-Task9-Task2-Task8-end con un tiempo de 15 unidades