

Programación 3

Trabajo Práctico Especial



Cursada 2024

Contexto

El trabajo práctico de cursada propone resolver distintos problemas asociados a un contexto simplificado de procesadores y tareas.

En una primera etapa se deberán resolver servicios simples de búsqueda sobre las tareas, mientras que en la segunda etapa el objetivo estará centrado en resolver asignaciones de tareas a procesadores, utilizando distintas técnicas algorítmicas.

Para ambas etapas el conjunto de datos de entrada estará dado por 2 archivos de texto con un formato predefinido, que se menciona a continuación.

Procesadores.csv

<id_procesador>;<codigo_procesador>;<esta_refrigerado?>;<año_funcionamiento>

Tareas.csv

<id_tarea>;<nombre_tarea>;<tiempo_ejecucion>;<es_critica?>;<nivel_prioridad>

Como se puede ver una tarea estará determinada por un ID, un nombre, un tiempo de ejecución, si es crítica o no y un nivel de prioridad (1 a 100).

Mientras que los procesadores se definen por un ID, un código, si están refrigerados o no y el año de puesta en funcionamiento.

El primer problema a resolver es implementar un algoritmo que reciba el nombre del archivo .csv de procesadores y el nombre del archivo .csv de tareas, procese ambos archivos y la información obtenida, para que luego pueda ser utilizada por los distintos algoritmos a implementar.

Primera parte

El objetivo de la primera parte es plantear una resolución eficiente para los siguientes servicios:

Servicio 1: Dado un identificador de tarea obtener toda la información de la tarea asociada.

Servicio 2: Permitir que el usuario decida si quiere ver todas las tareas críticas o no críticas y generar el listado apropiado resultante.

Servicio 3: Obtener todas las tareas entre 2 niveles de prioridad indicados.

Se establece la siguiente clase para la resolución de los servicios, la cuál deberá ser utilizada sin modificar la interfaz de los métodos públicos pero pudiendo agregar los métodos y estructuras privadas que consideren apropiadas.

```
public class Servicios {
    //Completar con las estructuras y métodos privados que se
    requieran.

    /*
     * Expresar la complejidad temporal del constructor.
     */
    public Servicios(String pathProcesadores, String pathTareas) {

    }

    /*
     * Expresar la complejidad temporal del servicio 1.
     */
    public Tarea servicio1(String ID) {    }

    /*
     * Expresar la complejidad temporal del servicio 2.
     */
    public List<Tarea> servicio2(boolean esCritica) {}

    /*
     * Expresar la complejidad temporal del servicio 3.
     */
    public List<Tarea> servicio3(int prioridadInferior, int
    prioridadSuperior) {}
}
```

Segunda parte

Ahora se desea establecer una asignación de todas las tareas a los distintos procesadores con el objetivo de minimizar el tiempo final de ejecución. Establecemos el tiempo final de ejecución como el tiempo máximo de ejecución de un procesador una vez que todas las tareas fueron asignadas.

Además, se sabe que tenemos ciertas restricciones para asignar una tarea a un procesador:

- Primero, ningún procesador podrá ejecutar más de 2 tareas críticas.
- Segundo, los procesadores no refrigerados no podrán dedicar más de X tiempo de ejecución a las tareas asignadas. El tiempo X será un parámetro establecido por el usuario al momento de solicitar la asignación de las tareas a los procesadores.

Objetivo

El objetivo de esta segunda parte del trabajo será resolver el problema planteado mediante dos técnicas algorítmicas distintas: Backtracking y Greedy.

Luego se deberán presentar los resultados teniendo en cuenta distintas métricas que permitan visualizar, mínimamente, la calidad de la solución y el costo de obtener dicha solución, con ambas técnicas.

Para la presentación de resultados se propone el siguiente formato:

Backtracking

Solución obtenida: cada procesador con las tareas asignadas.

Solución obtenida: tiempo máximo de ejecución.

Métrica para analizar el costo de la solución (cantidad de estados generados)

Greedy

Solución obtenida: cada procesador con las tareas asignadas.

Solución obtenida: tiempo máximo de ejecución.

Métrica para analizar el costo de la solución (cantidad de candidatos considerados)

También se solicita una breve explicación de la estrategia elegida para resolver el problema, mediante cada técnica solicitada. Para esto, se propone acompañar la implementación de la técnica con un breve comentario antes de la función principal, por ejemplo:

```
/*  
 * <<Breve explicación de la estrategia de resolución>>  
 */  
public Solucion backtracking(...) {}
```

Requisitos de la entrega

Se deberá entregar un proyecto (junto al código fuente) que compile correctamente el código de la aplicación solicitada. La entrega se realizará por email a la cuenta de la materia, pudiendo adjuntar en un archivo comprimido sólo el código fuente o compartiendo un link a un repositorio de git.

La entrega se realizará el **viernes 7/6**. En las semanas posteriores se realizará una devolución del trabajo entregado permitiendo, de ser necesaria, una re-entrega con las correcciones requeridas.

Importante: no se aceptarán entregas fuera de término.