**Medición de algoritmo implementado en diferentes estrategias de diseño de algoritmos**

***Objetivos***Desarrollar un programa en el lenguaje java utilizado las ventajas de OO.Implementar diferentes estrategias de diseño para la resolución de algoritmos

Realizar mediciones empíricas y analíticas para determinar eficiencia según la implementación del algoritmo.

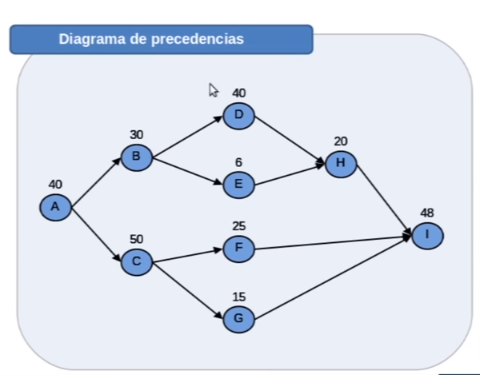
***Definición***

Resolver el problema de equilibrio de la línea de ensamblaje (ALBP= Assembly line balancing problem). El cual es un problema de optimización que busca el equilibrio de la línea de ensamblaje. Se da un conjunto de tareas (para realizar un producto), cada una de las cuales tiene un tiempo determinista y los predecesores respectivos, por lo cual las tareas deben llevar un orden de precedencia.

**Ejemplo**:

****

**Grafo de precedencia**

****

El balanceo de la línea de ensamble consiste en asignar todas las tareas a una serie de estaciones de trabajo de modo que cada una de ellas no tenga más de lo que puede hacer en el tiempo de ciclo de dicha estación y que el tiempo inactivo de todas las estaciones de trabajo sea el menor posible. [Ver](https://www.gestiondeoperaciones.net/procesos/ejemplo-del-balance-de-una-linea-de-ensamble-utilizando-la-regla-del-candidato-mas-extenso/) [Ver](https://youtu.be/kWsOQsxE0W8)

Las estrategias de diseño de algoritmos a desarrollar son:

**Programación dinámica, Algoritmo Genético.**

***Mediciones sobre los algoritmos***

Debe realizar pruebas para medir las dos diferentes estrategias de diseño, probando **con grandes tamaños de tareas diferentes**. Con el mismo ejemplar de grafo se aplican las 2 estrategias diseño. Para determinar el factor de talla debe realizar 5 comparaciones entre los diferentes tamaños.

Se le solicita realizar el conteo la cantidad de memoria consumida (contar las variables que se utilice por ejemplo si declara un int32 suma 32 bits. Si declara un char suma 8 bits) en cada estrategia, así como las comparaciones y asignaciones determinar el factor de crecimiento y clasificar los algoritmos en notación O.

Además, debe realizar la medición analítica para cada uno de los algoritmos de las estrategias de diseño.

Debe usar tablas similares a las que se usaron para el primer proyecto programado.

Los ejemplares pequeños deben venir en el código (quemados).

Tamaños de grafo:

1. 10 tareas.
2. 20 tareas.
3. 30 tareas.
4. 60 tareas.
5. 120 tareas de aquí en adelante puede generar aleatorios.
6. 1200 tareas.

***Consultas***

1. Imprima todas las variables de medición (memoria, asig, comp) antes mencionadas para cada una de las estrategias de diseño.
2. Imprimir la respuesta de cada algoritmo, tareas asignadas por estación, tiempo en ejecución, tiempo ocioso….
3. Imprimir todos los cruces realizados para la estrategia genética y su mutación. Así como la puntación asignada a cada cromosoma.
4. Para la estrategia de programación dinámica imprimir algunas (5) fases durante el proceso.

***Nota:*** Tomar en cuenta todas las aclaraciones que se dieron en día de entrega y discusión de este proyecto. Para más información consultar

### Documentación Externa

Portada,

Introducción

Análisis del problema, cite las referencias usadas.

Solución del problema

* Hacer un diagrama con las estructuras utilizadas, listas, pilas, arreglos u otras estructuras, por estrategia.
* Realice un diagrama de flujo o seudocódigo donde explique la lógica que desarrollo, para aplicar cada una de las estrategias
* Describa cuál es la estrategia seleccionada en programación dinámica para avanzar entre las fases.
* Describa cuáles son los tipos de cruce realizados.
* Describa cuál es el tipo de mutación que se aplicó.

Análisis de Resultados

* Resultados finales, indique que partes están completas, cuales defectuosos, y cuáles no se realizaron y el porqué.
* Adjunte las tablas de todas las mediciones realizadas a sus algoritmos. Junto con el dato de prueba más pequeño.

Conclusiones

* Según la medición realizada indique cuál estrategia de diseño es más eficiente, justifique su aseveración.
* **Responda la siguiente pregunta ¿Conforme crece la talla cual algoritmo se va haciendo más eficiente?**

Recomendaciones

* Con respecto al alcance del proyecto.

Literatura citada

* Mínimo de debe incluir 4 con su respectivo resumen. Use el formato APA.

### Documentación Interna

Fecha de inicio y Fecha última modificación.

Descripción para cada estructura (o clase) y su uso en el programa,

Describir cada función e instrucciones dentro de estas.

### Aspectos Administrativos

* La tarea debe programarse en lenguaje java
* El desarrollo de este trabajo se puede realizar en pareja como máximo.
* Entrega de la tarea final: 13 de noviembre antes de las 11 p.m.
* Se calificará con citas de revisión para la defensa de la tarea de ser necesario.
* Si se encuentra copia la calificación será de cero para todos los implicados. No subir partes o avances del proyecto en repositorios públicos antes de la revisión del proyecto.
* Debe entregarse la tarea en el TecDigital. Si tiene virus o si se encuentra mal identificando se rebajarán puntos por descuido del estudiante. Si no abre el proyecto no se calificará la parte programada.

Se recomienda que se comience a trabajar desde hoy.