**Estrategias de diseño**

***Objetivos***

* Desarrollar un programa en el lenguaje java utilizado las ventajas de OO.
* Implementar tres estrategias de diseño diferentes para la resolución de problemas.
* Realizar mediciones empíricas y analíticas para determinar eficiencia según la implementación del algoritmo.

***Definición*** Las empresas, organizaciones, centros de estudio entre otros; cada día deben ser más eficientes; por lo que deben tener una asignación de recursos óptima. Por ejemplo: la asignación de tareas a los trabajadores. Cada trabajador puede realizar una tarea mejor que otra. Así que si se logra asignar la tarea que mejor realiza a cada trabajador se obtendrá una optimización global.

Por ejemplo, si en una universidad hay dos profesores que ofrecen el curso de Compiladores y el curso de Arquitectura y se tienen los siguientes datos históricos de calificación (ver tabla), ¿Cuál sería la mejor asignación de cursos a los profesores; si la universidad desea ofrecer la mejor calidad de educación y servicios para los estudiantes?

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre**/**curso** | Compiladores | Arquitectura | .. | m |
| Marco | 90 | 62 |  |  |
| Roberto | 70 | 88 |  |  |
| Miguel | - | - |  |  |
| … |  |  |  |  |
| n |  |  |  |  |

Respuesta:

|  |  |  |
| --- | --- | --- |
| **Nombre**/**curso** | Compiladores | Arquitectura |
| Marco | **x** |  |
| Roberto |  | **x** |

Los datos de entrada se tienen que realizar quemados en el código para los siguientes tamaños:

|  |
| --- |
| **Tamaño x**  (prof/cursos) |
| 3p, 9c |
| 5p, 15c |
| 6p, 18c |
| 9p, 27c |
| 10p, 20c |
| 10p, 30c |

Nota: las calificaciones van de 0 a 100, -1 para indicar que ese profesor no da ese curso.

Para encontrar la mejor respuesta debe utilizar tres estrategias de diseño:

algoritmos genéticos, programación dinámica y backtracking.

**Restricciones del problema**

* Debe considerar que la carga de un profesor o trabajador no debe exceder de un tiempo semanal, el cual representa de 3 a 4 cursos.
* Todos los cursos disponibles se deben asignar una única vez.
* No pueden quedar cursos sin asignar.
* Podría algún trabajador quedar sin asignación.

***Algoritmo Genético:***

***Población inicial***

El programa creará la población inicial de forma aleatoria o usando un algoritmo voraz, cumpliendo con las restricciones del problema. El tamaño máximo de cada individuo o cromosoma es x. Cada gen está compuesto por 3 alelos (profesor, curso y calificación).

*Cantidad de población por tamaño*

|  |  |
| --- | --- |
| **Tamaño x**  (prof/cursos) | **Cantidad**  **población** |
| 3p, 9c | 3 |
| 5p, 15c | 5 |
| 6p, 18c | 10 |
| 9p, 27c | 20 |
| 10p, 20c | 20 |
| 10p, 30c | 20 |

***Función aptitud o fitness***

El objetivo de este algoritmo es encontrar la asignación de cursos a profesores que optimice una función de aptitud: maximizar la calidad en la asignación de cursos.

Cada población generada se debe evaluar, cuando ocurre una generación las poblaciones compiten entre sí y con sus padres. Solo quedaran las mejores poblaciones evaluadas, siempre manteniéndose el número de la población indicada en la tabla anterior.

La evaluación debe ser un valor numérico.

***Cruces***

Se seleccionan dos cromosomas padres y se realiza el cruce entre ellos para generar dos cromosomas hijos, que heredan información de ambos padres.

El programa tomara la población generada y evaluada de mayor a menor para realizar un tipo de cruce que no genere alelos repetidos y generar la cantidad de hijos indicados en la siguiente tabla:

|  |  |  |
| --- | --- | --- |
| **Tamaño x**  (prof/cursos) | **Cantidad**  **población** | **Cantidad**  **de hijos** |
| 3p, 9c | 3 | 6 |
| 5p, 15c | 5 | 10 |
| 6p, 18c | 10 | 20 |
| 9p, 27c | 20 | 40 |
| 10p, 20c | 20 | 40 |
| 10p, 30c | 20 | 40 |

Cuando se terminen las generaciones solicitadas como se indica en la siguiente tabla, se deben obtener los 5 mejores resultados para evaluar su efectividad en la respuesta aproximada u óptima.

***Mutación***

En caso de poblaciones repetidas aplicar un tipo de mutación, en caso de mejorar la puntuación se queda aplicada, en caso de empeoramiento en la evaluación descartarla. Igual se debe cumplir con las restricciones del problema.

***Generaciones***

Cada vez que se cruza toda la población evaluada es una generación. El programa deberá realizar un ciclo para generar varias generaciones como se indica en la siguiente tabla:

|  |  |  |  |
| --- | --- | --- | --- |
| **Tamaño x**  (prof/cursos) | **Cantidad**  **población** | **Cantidad**  **de hijos** | **Cantidad de**  **Generaciones** |
| 3p, 9c | 3 | 6 | 3 |
| 5p, 15c | 5 | 10 | 5 |
| 6p, 18c | 10 | 20 | 10 |
| 9p, 27c | 20 | 40 | 20 |
| 10p, 20c | 20 | 40 | 20 |
| 10p, 30c | 20 | 40 | 20 |

Reemplazo: Se seleccionan los cromosomas más aptos para sobrevivir a la siguiente generación y se descartan los menos aptos, siempre manteniendo la misma cantidad de población.

Al finalizar se deben comparar los mejores 5 resultados para determinar la mejor respuesta.

***Mediciones sobre los algoritmos***

Debe realizar pruebas para medir los tres algoritmos probando con los tamaños indicados. Para determinar el factor de talla debe usar todos los tamaños y realizar al menos 7 cálculos diferentes.

Debe realizar el conteo de asignaciones, comparaciones, tiempo y memoria consumida.

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. Debe determinar el factor de crecimiento y clasificar los algoritmos en notación O.

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

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

***Consultas***

1. Imprimir la mejor combinación obtenida con su calificación total.

Ejemplo: Marco – Compiladores.

Roberto- Arquitectura.

Calificación = 178

1. Imprima todas las variables de medición (memoria, tiempo, asignaciones, comparaciones y cantidad de total de instrucciones) para cada algoritmo.
2. Imprimir todos los cruces realizados en la estrategia genética. Así como la puntación asignada a cada cromosoma.

Padre1 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_puntuación \_\_\_\_\_\_

Padre 2\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ puntuación \_\_\_\_\_\_

Hijo 1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ puntuación \_\_\_\_\_\_

Hijo 2\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_puntuación \_\_\_\_\_\_

1. Imprimir las mutaciones aplicadas. Así como la puntación asignada a cada cromosoma.

Individuo 1 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_puntuación \_\_\_\_\_\_

Mutación \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_puntuación \_\_\_\_\_\_

Individuo 2\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ puntuación \_\_\_\_\_\_

Mutación \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_puntuación \_\_\_\_\_\_

1. Imprimir las 5 mejores poblaciones con su puntación por cada tamaño, al finalizar la cantidad de generaciones solicitadas.
2. En la ejecución del backtracking indique la cantidad total de combinaciones completas se generaron, cumpliendo con todas las restricciones del problema.
3. En la ejecución de la programación dinámica imprima los resultados previos cada 5 etapas (inicial, la 5, 10, 15, 20 ….).

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

### Documentación Externa

Portada,

Introducción

Análisis del problema.

Solución del problema

* Indique en que estructura de datos almacena los datos quemados, que datos tiene, cuales datos definió de control para cumplir con las restricciones del problema.
* Hacer un diagrama con las estructuras utilizadas, listas, pilas, arreglos u otras estructuras, por estrategia de diseño.
* Describa cuál es el cruce realizado.
* Realice un diagrama de flujo o seudocódigo donde explique la lógica que desarrollo, para aplicar cada uno de los algoritmos.
* Describa cuál es el tipo de mutación que se aplicó.
* Describa la función aptitud, cálculo matemático
* Describa la estrategia de la programación dinámica, si los datos se ordenan por curso o profesor de forma descendente o ascendente, que criterio se utiliza para la selección o asignación de cada curso en cada etapa de avance…
* Describa la estrategia del backtracking, como va construyendo la respuesta en cada llamada recursiva, como descarta las posibilidades que no cumplen con las restricciones del problema. Para que tamaños de problemas es posible dar respuesta.

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.
* Cálculos realizados.
* Clasificación en notación O grande.
* Gráfico y su respectivo análisis donde compare los comportamientos de los algoritmos.
* Análisis de las tablas de mediciones (empíricas y analíticas).

Conclusiones

* Según la medición realizada indique cuál estrategia es más eficiente para resolver este problema, justifique su aseveración.
* Responda la siguiente pregunta ¿Conforme crece la talla cuál 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.

Bitácora y minutas

### 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, NetBeans.
* El desarrollo de este trabajo se puede realizar grupos de tres.
* Entrega final de la tarea es el 7 de junio antes de las 11 p.m.
* Si se encuentra copia la calificación será de cero para todos los implicados.
* 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.