Análisis de Algoritmos
Profesora Ana Lorena Valerio Solís
Proyecto programado # 2
I semestre 2024

Estrategias de diseño

Objetivos

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

Definición

Sabemos que una estrategia no es mejor que otra por si sola, depende del problema resolver. En este proyecto se quiere resolver un problema clásico de la mochila aplicando dos estrategias de diseño diferentes: programación dinámica y genética.
Se debe construir una única representación de los objetos, cada uno con su peso y su valor, y la capacidad máxima de la mochila. Las 2 estrategias, debe tomar de referencia la representación única. Por tanto, todo debe programarse en un solo proyecto.

Los datos de entrada son los siguientes tamaños:

Cantidad de objetos
5
10
20
30
40
50

Para el tamaño 5.

Objeto	Bloqueador	Ropa	Sombrilla	Desodorante	Cepillo de dientes
Valor	9	10	7	9	10
Peso	3	5	2	3	2

Para los tamaños de 10 en adelante, los pesos y valor se asignarán con un valor random de 1 a 50.
El programa debe imprimir la respuesta (los objetos seleccionados con su nombre, valor y peso) e indicar el total de peso y el total de valor, para cada una de las estrategias.

Restricciones del problema

Ninguna respuesta puede superar la capacidad máxima de la mochila.
Ningún valor puede ser negativo.
Ningún peso puede ser negativo.

Análisis de Algoritmos
Profesora Ana Lorena Valerio Solís
Proyecto programado # 2
I semestre 2024

Programación dinámica

Crear una tabla de las siguientes dimensiones la cantidad de filas sería la cantidad de objetos, la cantidad de columnas sería del valor de la capacidad máxima de la mochila. Cada casilla se va llenando iterativamente de forma ascendente, el valor de la celda almacenará el valor máximo alcanzado hasta ese momento. Al finalizar el recorrido, la ultima celda tiene el máximo.
Debe ir almacenando los objetos que se van seleccionado para imprimirlos al final como respuesta, y con el máximo alcanzado.

Estrategia Genética

Población inicial

El programa creará la población inicial de forma aleatoria o usando un algoritmo voraz, cumpliendo con las restricciones del problema. No puede existir población repetida.

Cantidad de objetos	Cantidad de población inicial
5	3
10	10
20	20
30	30
40	40
50	50

Cromosoma

Está compuesto por genes. Cada gen representa la presencia o ausencia de un objeto en la mochila.

Ejemplo #1 de cromosoma de tamaño 5

Objeto 1	Objeto 2	Objeto 3	Objeto 4	Objeto 5
1	0	0	1	0

Ejemplo #2 de cromosoma de tamaño 5

Objeto 1	Objeto 2	Objeto 3	Objeto 4	Objeto 5
0	1	0	1	1

Función aptitud o fitness

El objetivo de este algoritmo es encontrar la combinación de objetos que tenga mayor valor.
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, en este caso el valor total de la combinación de objetos presentes.

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:

Cantidad de objetos	Cantidad de población inicial	Cantidad de hijos
5	3	6
10	10	20
20	20	40
30	30	60
40	40	80
50	50	100

Generar 20 generaciones, se deben obtener los 5 mejores resultados para evaluar su efectividad en la respuesta aproximada u óptima.

Mutación

En caso de poblaciones repetidas o que superar la capacidad de la mochila 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 20 generaciones para todos los tamaños.

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 dos 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 respuesta (objetos con nombre, peso y valor ind) con el valor y peso totales.
2. Imprima todas las variables de medición (memoria, tiempo, asignaciones, comparaciones y cantidad de total de instrucciones) para cada algoritmo.
3. 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 _____

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

Individuo 1 _____puntuación _____
Mutación _____puntuación _____

Individuo 2_____ puntuación _____
Mutación _____puntuación _____

5. Imprimir las 5 mejores poblaciones con su puntación por cada tamaño, al finalizar la cantidad de 20 generaciones.
6. En la ejecución de la programación dinámica imprima los resultados previos cada 5 etapas o cada 5 objetos analizados (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, buscar alternativas de solución, que complejidad computacional tienen, que ventajas y desventajas tienen con respecto a la cantidad de datos.
Solución del problema

- Indique en que estructura de datos almacena los datos quemados, que atributos tiene, cuales datos definió de control para cumplir con las restricciones del problema y el algoritmo propuesto.
- 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.

Análisis de Algoritmos
Profesora Ana Lorena Valerio Solís
Proyecto programado # 2
I semestre 2024

- Describa la estrategia de programación dinámica, como avanza en las diferentes etapas, cuando inicia con la solución vacía como va agregando objetos a las posible respuesta. Donde almacena los resultados previos, para los siguientes cálculos como se basa en el resultado previo para calcular el siguiente.

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.
- Primer gráfico de las instrucciones ejecutadas y su respectivo análisis donde compare los comportamientos de los algoritmos. En el eje X los tamaños de objetos, en el eje Y la cantidad de instrucciones
- Segundo gráfico con la memoria consumida. En el eje X los tamaños de objetos, en el eje Y la memoria consumida.
- 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.
- El desarrollo de este trabajo se puede realizar grupos de tres.
- Entrega final de la tarea es el 4 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.