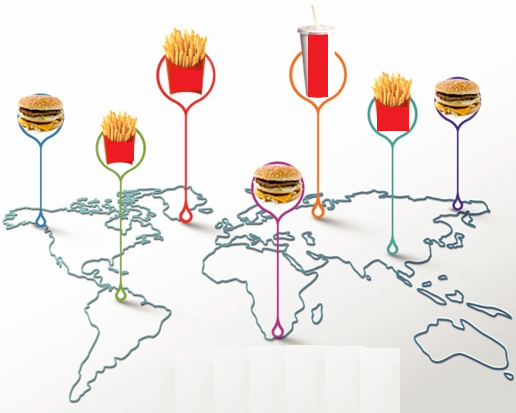
**Análisis de Algoritmos**

**Profesora Ana Lorena Valerio SolísProyecto programado # 1**

**Análisis de complejidad algorítmica para el problema de ubicación de restaurantes de una cadena comercial**

**Objetivos**

* Implementar algoritmos de maximización de rendimientos en un grafo de forma eficiente en el lenguaje Java, utilizado las ventajas de OO
* Analizar los algoritmos mediante mediciones empíricas y analíticas.

**Definición del problema**

Una cadena de restaurantes, quiere construir restaurantes en muchos sitios con el objetivo de maximizar su beneficio total.

La red (grafo) de calles se describe como un gráfico no dirigido G = (V, E), donde los sitios de los potenciales restaurantes son los vértices del grafo. Cada vértice *u* tiene un valor entero no negativo *pu*, que describe el beneficio potencial del sitio *u*. Como restricción del problema, no se pueden construir dos restaurantes en vértices adyacentes (para evitar la auto-competencia). Debe diseñar un algoritmo que genere el conjunto elegido U ⊆ V de sitios que maximiza el beneficio total u∈*U* pu.

Para las siguientes partes (a) - (b) - (c), suponga que la red (grafo) es acíclica, es decir, un árbol.

1. Realice el siguiente algoritmo voraz (greedy) de colocación de restaurantes: Elija el vértice con el más alto beneficio u0 en el árbol y póngalo en U. Elimine u0 de mayor consideración, junto con todos sus vecinos en G. Repita hasta que no queden más vértices.

Dé un contraejemplo para mostrar que este algoritmo no siempre coloca un restaurante con el máximo beneficio esperado (en la documentación externa).

1. Realice un algoritmo eficiente para determinar la ubicación con el máximo beneficio.
2. Suponga que, en ausencia de una buena investigación de mercado, la cadena de restaurantes decide que todos los sitios son igualmente buenos, por lo que el objetivo es realizar un algoritmo para la ubicación de restaurantes con el mayor número de ubicaciones. Realice un algoritmo voraz para este caso y pruebe su correcta ejecución.
3. Ahora suponga que el grafo es arbitrario, no necesariamente acíclico. Desarrolle el algoritmo correcto más rápido que pueda para resolver el problema.

**Mediciones sobre los algoritmos.**

Para evaluar la efectividad y eficiencia de los algoritmos, se implementarán pruebas y mediciones para cada uno de los cuatro algoritmos, con diferente cantidad de datos de entrada por defecto, para determinar si los algoritmos son tratables o intratables.

A continuación, una descripción de los procesos de medición y registro de datos que se implementarán para cada uno de los algoritmos:

1. Medición empírica, estadísticas básicas

* Tiempo de ejecución en segundos/milisegundos. Recuerden utilizar una precisión de 3 decimales.
* Comparaciones (menor “<”, mayor “>”, igual que “==”, distinto de “!=”) realizadas en tiempo de ejecución.
* Asignaciones realizadas en tiempo de ejecución.
* Cantidad de líneas del código que componen los algoritmos
* Cantidad total de líneas ejecutadas en cada prueba

1. Cálculo del factor de crecimiento y clasificación de los algoritmos en notación: Theta, O Grande, y Omega.
2. Clasificación en notación O Grande ***según sus comparaciones y asignaciones***.
3. Medición analítica de cada uno de los códigos de los algoritmos y determinar su clasificación en notación O Grande.

Para registrar los datos anteriores, se adjuntan las siguientes tablas.

Medición empírica

* 1. Nombre del algoritmo #a: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

| Operaciones | Cantidad de datos de entrada por defecto | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| Asignaciones |  |  |  |  |  |  |
| Comparaciones |  |  |  |  |  |  |
| Cantidad de líneas ejecutadas |  |  |  |  |  |  |
| Tiempo de ejecución |  |  |  |  |  |  |
| Cantidad de líneas del código |  | | | | | | |

Crear una tabla igual para los otros algoritmos.

Determinar el factor de crecimiento

3.1 Nombre del algoritmo #a: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

| Talla | | Factor talla | Factor Asig | Factor Comp | Factor Cantidad de líneas ejecutadas | Factor Tiempo de ejecución |
| --- | --- | --- | --- | --- | --- | --- |
| De -- a -- |  |  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| De -- a -- |  |  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| De -- a -- |  |  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| De -- a --- |  |  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| De -- a --- |  |  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| De --- a ---- |  |  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| Clasificación del comportamiento de las **asignaciones** |  |  |
| Clasificación del comportamiento de las **comparaciones** |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Clasificación según su entrada de los datos use la notación  Theta, O Grande, y Omega según corresponda | | | |
| **Entrada de los datos** |  |  |  |
| Clasificación |  |  |  |

Crear una tabla igual para los otros algoritmos.

Medición analítica

* 1. Nombre del algoritmo de #a: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |
| --- | --- |
| Código fuente  Solo se analiza el código del método de resolver el rompecabezas. | Medición de líneas ejecutadas en el peor de los casos  (línea por línea) |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| Total (la suma de todos los pasos) |  |
| Clasificación en notación O Grande |  |

***Crear una tabla igual para los otros algoritmos.***

Medición gráfica:

### Grafique el comportamiento de los algoritmos, en un solo gráfico, tomando en cuenta solo la variable de comparaciones y en un segundo gráfico las asignaciones. Para cada algoritmo obtenga la medición de 10 pares ordenados, a una escala pequeña. Recuerde indicar el nombre al gráfico, nombre a los ejes y serie clara para cada una de las gráficas.

### Realice un análisis del gráfico, compárelo con el comportamiento de la clasificación asignada a los algoritmos en la medición empírica.

**Operaciones**

Crear los algoritmos correctamente de forma eficiente y presentar en consola las respuestas para los tamaños que sean posibles. TODOS en un solo proyecto, no se revisará más de un proyecto.

**A la hora de ejecutarlo el usuario final no introducirá ningún dato.** Los datos de entrada serán por defecto.

Mostrar en pantalla los resultados de medición (calculados correctamente) cuando el tamaño de la prueba lo permita.

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

**La documentación Externa debe contener**

**Portada.**

**Introducción**.

**Análisis del problema**.

*[ Debe analizar todos los puntos a tratar en el proyecto, considerando estos como requerimientos de un proyecto de software o como las preguntas o situaciones a resolver en un proceso de indagatoria (investigación). Incluir una descripción eficiente de la situación y las metodologías o medios necesarios para brindar una posible solución o abordaje al problema.]*

**Solución del problema**

* la última solución, indique cuales son las estructuras o clases utilizadas.
* lógica de cómo se realizó cada algoritmo, indique todas las mejoras realizadas a los algoritmos para mejorar su eficiencia.

**Análisis de Resultados.**

* Resultados finales, indique que partes están completas, cuales defectuosos, y cuáles no se realizaron y el porqué, que aspectos se pueden mejorar.
* Adjunte las tablas de todas las mediciones realizadas a sus algoritmos (empírica, analítica y la medición gráfica).
* Adjunte el cálculo del factor de crecimiento.
* Analice los datos obtenidos, a que se debe que un algoritmo sea mejor que otro. Indique las características o ventajas de cada algoritmo sobre el otro.

**Conclusiones**

* Según la medición realizada indique cuál de los algoritmos es más eficiente analíticamente y empíricamente; tanto en comparaciones y asignaciones y según sus la entrada de los datos.

**Recomendaciones**

* Aspectos o ámbitos de interés para ampliar el alcance del proyecto.
* Áreas o campos de estudio que pueden complementar las temáticas analizadas.
* Ideas que sean de utilidad para interesados en el área en que se desenvuelve el proyecto.

**Literatura citada**

* Mínimo de debe incluir 4 referencias, use el formato APA, incluya las consultas realizadas en la web.

### Documentación Interna

Fecha de inicio y Fecha última modificación.

Usar la Notación CamelCase:

* + - LowerCamelCase: Cada variable deberá definirse con su primera palabra en minúscula y las siguientes con su primera letra en mayúsculas. Ejemplos:
      * primerNombre, numeroPuestos, correoPersonal.

Descripción para cada 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.
* La fecha de entrega de la tarea: 28 de septiembre de 2019 antes de las 11:30 p.m.
* Si se encuentra copia en alguna parte del proyecto la calificación será de cero para todos los implicados.
* Debe entregarse la tarea a través del tec-Digital, si tiene virus o si se encuentra mal identificando se rebajarán puntos por descuido del estudiante. Si no abre el proyecto o la documentación, se calificará con una nota de 0.
* Se recomienda que se comience a trabajar desde hoy. Recuerde existen horas de consulta.
* Las mediciones en tiempo de ejecución deben coincidir con las presentadas en la documentación, en caso contrario se rebajarán los puntos correspondientes.

### Calificación General.

60% Documentación que incluye las diferentes mediciones.

40% Programación.

* Se rebajarán puntos por utilizar código o programación poco eficiente, redundante o innecesaria. Por ejemplo: declaración de variables sin usar, o de un tipo de datos incorrecto, en vez de un boolean un char o string.