**Análisis de Algoritmos**

**Grupo 50**

**Profesora**

Ana Lorena Valerio Solís

**Instituto Tecnológico de Costa Rica**

**Proyecto programado # 1**

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

**Integrantes**

Nelson Andrey Vega Soto

Yuliana Rojas

**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.

**Nombre del algoritmo A: Greedy**

**Medición Empírica**

| Operaciones | Cantidad de datos de entrada por defecto | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| 6 | 10 | 15 | 20 | 25 | - |
| Asignaciones | 47 | 109 | 184 | 340 | 505 | - |
| Comparaciones | 53 | 156 | 271 | 554 | 858 | - |
| Cantidad de líneas ejecutadas | 100 | 265 | 455 | 894 | 1363 | - |
| Cantidad de líneas del código | 12 | | | | | |

**Factor de Crecimiento**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Talla | Factor talla | Factor Asig | Factor Comp | Factor Cantidad de líneas ejecutadas |
| De 6 a 10 | 1.6 | 2.25 | 2.94 | 2.56 |
| De 10 a 15 | 1.5 | **1.68** | **1.73** | **1.77** |
| De 15 a 20 | 1.3 | 1.84 | 2.04 | 1.93 |
| De 20 a 25 | 1.2 | 1.48 | 1.54 | 1.52 |
| De 15 a 25 | 1.6 | 2.74 | 3.16 | 2.99 |
| De 10 a 20 | 2 | 3.11 | 3.55 | 3.37 |

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

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

**Factor de Crecimiento**

|  |  |
| --- | --- |
| 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) |
| public void greedy(Grafo grafoM){  ArrayList<Vertice> grafo = grafoM.grafo;  int beneficioTotal = 0;  for(int i=0 ; i< grafo.size() ; i++){  if(!grafo.get(i).visitado){  grafo.get(i).visitado = true;  beneficioTotal += grafo.get(i).getBeneficio();  for(int j=0 ; j<grafo.get(i).listaArcos.size() ; j++){  grafoM.buscarVertice(grafo.get(i).listaArcos.get(j).destino).visitado = true;  }  }  }  } | 1  1  1 + 1+n + n  n  n  n  n (1+1+n+n)  1 |
|  |  |
|  |  |
|  |  |
|  |  |
| Total (la suma de todos los pasos) | 4 + 7n + 2n2 |
| Clasificación en notación O Grande | O(2n2) cuadrática |

**Nombre del algoritmo #2:**

**Medición Empírica**

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

**Factor de Crecimiento**

| 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 |  |  |  |

**Factor de Crecimiento**

|  |  |
| --- | --- |
| 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 |  |

**Nombre del algoritmo #3:**

**Medición Empírica**

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

**Factor de Crecimiento**

| 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 |  |  |  |

**Factor de Crecimiento**

|  |  |
| --- | --- |
| 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 |  |

**Nombre del algoritmo #4:**

**Medición Empírica**

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

**Factor de Crecimiento**

| 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 |  |  |  |

**Factor de Crecimiento**

|  |  |
| --- | --- |
| 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 |  |

**Conclusiones:**

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