**ACTIVIDAD 15: ANÁLISIS EMPÍRICO**

SECCIÓN 1: ANALISIS TEORICO ESPACIO-TEMPORAL

Sección 1.1: Porqué Lista de Adyacencias y no Matriz de Adyacencias?

Una de las primeras decisiones que fue necesario tomar para afrontar esta actividad fue la selección de la implementación de la ED Grafo más eficiente para el problema, luego de deliberar sobre la cuestión llegamos a la conclusión que la mejor implementación en este caso sería el Grafo con listas de Adyacencias, veamos el porqué

Sabemos por lo visto en teoría que el espacio de ejecución de la implementación por matriz de Adyacencia es O(n^2), mientras que el espacio para la lista de adyacencias es O(n + a), por las restricciones del problema, en el peor caso (donde n = 500 y a = (500\*499)/2, el espacio para la matriz de adyacencias seria O(500^2) = 250000 mientras que el espacio en la lista de adyacencias seria O(500 + 124750) = 125250 , la diferencia de espacio es aproximadamente la mitad y entonces se justifica el uso de la Lista de Adyacencias

Y que ocurre con el tiempo de ejecución? Sabemos que, en este aspecto, la Matriz de Adyacencias es mejor ya que al ser una matriz, los accesos a la misma se pueden realizar en tiempo constante mientras que los accesos en la lista de adyacencias son O(n+a), pero en este aspecto, es posible reducir el tiempo de los accesos a las listas por medio del uso de punteros

Sección 1.2: Análisis de Tiempo y Espacio de las EDs:

Analicemos los tiempos y espacios de Ejecución ED por ED:

Cola: Esta ED se comporta de la misma forma que la ED Cola vista en clase, por ende, los tiempos de las operaciones son:

Grafo: Esta ED implementa un Grafo según la implem

Disjoint-Set:

Heap:

Sección 1.3: Análisis de Tiempo y Espacio de los Problemas:

Analicemos los tiempos y espacios de ejecución problema por problema:

Problema 1: Grafo Conexo:

Problema A: BFS

Problema B: Conjunto Disjunto

Problema 2: Árbol Minimal de Cubrimiento:

Problema 1A: Kruskal Lista Ordenada con Disjoint Set Con Heurística

Problema 1B: Kruskal Lista Ordenada con Disjoint Set Sin Heurística

Problema 1C: Kruskal Heap con Disjoint Set Con Heurística

Problema 1D: Kruskal Heap con Disjoint Set Sin Heurística

SECCIÓN 2: TABLAS DE RESULTADOS EMPÍRICOS

Tabla de resultados empíricos para el ejercicio 1:

|  |  |  |  |
| --- | --- | --- | --- |
| Grafo | | BFS | Disjoint-Set |
| N | A |  |  |
| 500 | 40000 |  |  |
| 5 | 10 |  |  |
| 500 | 124750 |  |  |
| 200 | 15000 |  |  |
| 71 | 900 |  |  |
| 190 | 300 |  |  |
| 420 | 69870 |  |  |
| 2 | 1 |  |  |
| 179 | 179 |  |  |
| 50 | 49 |  |  |

Tabla de resultados empíricos para el ejercicio 2:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Grafo | | Ordenado | | Heap | |
| N | A | C/heurística | S/heurística | C/heurística | S/heurística |
| 80 | 100 |  |  |  |  |
| 500 | 124750 |  |  |  |  |
| 20 | 30 |  |  |  |  |
| 361 | 500 |  |  |  |  |
| 211 | 210 |  |  |  |  |
| 124 | 6999 |  |  |  |  |
| 2 | 1 |  |  |  |  |
| 173 | 10000 |  |  |  |  |
| 10 | 10 |  |  |  |  |
| 300 | 41258 |  |  |  |  |

SECCIÓN 3: CONCLUSIONES

SECCIÓN 4: CODIGOS FUENTE

El lenguaje elegido para implementar las soluciones y EDs fue Java, usando el IDE Eclipse