**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 por ende se elimina la ventaja que poseía la matriz en cuanto al orden de acceso a la estructura. Por ende, nos fue más valioso ahorrar en espacio de ejecución del grafo

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 implementación de Grafo con Listas de Adyacencias según la teoría, por ende, los tiempos de ejecución son:

Disjoint-Set: El Disjoint-Set esta presente de dos formas distintas, aunque ambas implementaciones respetan que la estructura interna de los cjtos es un árbol, las operaciones son implementadas de forma distinta, ya que una de las implementaciones del disjoint Set NO hace uso de las Heurísticas vistas en teoría, entonces, denominaremos al DIsjoint-Set SH a la implementación de disjoint set que NO hace uso de las heurísticas y a Disjoint-Set CH a la implementación de Disjoint-Set que HACE uso de las heurísticas vistas en teoría, entonces, los tiempos de ejecución son:

* DIsjoint-Set SH
  + Unión:
* DIsjoint-Set CH
  + Union

Heap: La ED Heap se comporta de la misma manera que la ED vista en teoría, entonces los timepos de ejecución son:

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

El problema fue resuelto siguiendo la siguiente estrategia: al realizar el recorrido BFS, ir marcando cada nodo visitado y una vez terminado el recorrido, verificar si con un solo BFS pude recorrer todo el grafo

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

A continuación se muestran las tablas de resultados empíricos producto de ejecutar los algoritmos implementados para una variedad de grafos, se resaltará en amarillo el tiempo menor entre las variantes de implementación para los problemas resueltos. Los tiempos se midieron en Nanosegundos.

Tengo el grafo en formato JSON. Lo convierto...

Grafo con 500 nodos y 40000 arcos construido

Tengo el grafo en formato JSON. Lo convierto...

Grafo con 5 nodos y 10 arcos construido

Tengo el grafo en formato JSON. Lo convierto...

Grafo con 500 nodos y 124750 arcos construido

Tengo el grafo en formato JSON. Lo convierto...

Grafo con 200 nodos y 15000 arcos construido

Tengo el grafo en formato JSON. Lo convierto...

Grafo con 71 nodos y 900 arcos construido

Tengo el grafo en formato JSON. Lo convierto...

Grafo con 190 nodos y 300 arcos construido

Tengo el grafo en formato JSON. Lo convierto...

Grafo con 420 nodos y 69870 arcos construido

Tengo el grafo en formato JSON. Lo convierto...

Grafo con 2 nodos y 1 arcos construido

Tengo el grafo en formato JSON. Lo convierto...

Grafo con 179 nodos y 179 arcos construido

Tengo el grafo en formato JSON. Lo convierto...

Grafo con 50 nodos y 49 arcos construido

Tengo el grafo en formato JSON. Lo convierto...

Grafo con 80 nodos y 100 arcos construido

Tengo el grafo en formato JSON. Lo convierto...

Grafo Conexo con 500 nodos y 124750 arcos construido

Tengo el grafo en formato JSON. Lo convierto...

Grafo Conexo con 20 nodos y 30 arcos construido

Tengo el grafo en formato JSON. Lo convierto...

Grafo Conexo con 361 nodos y 500 arcos construido

Tengo el grafo en formato JSON. Lo convierto...

Grafo Conexo con 211 nodos y 210 arcos construido

Tengo el grafo en formato JSON. Lo convierto...

Grafo Conexo con 124 nodos y 6999 arcos construido

Tengo el grafo en formato JSON. Lo convierto...

Grafo Conexo con 2 nodos y 1 arcos construido

Tengo el grafo en formato JSON. Lo convierto...

Grafo Conexo con 173 nodos y 10000 arcos construido

Tengo el grafo en formato JSON. Lo convierto...

Grafo Conexo con 10 nodos y 10 arcos construido

Tengo el grafo en formato JSON. Lo convierto...

Grafo Conexo con 300 nodos y 41258 arcos construido

el tiempo para el Problema 1 por BFS para el grafo1 es: 0.14493653

el tiempo para el Problema 1 por Disjoint-Set para el grafo1 es: 3.88558699

el tiempo para el Problema 1 por BFS para el grafo2 es: 0.017368639999999998

el tiempo para el Problema 1 por Disjoint-Set para el grafo2 es: 0.024202889999999998

el tiempo para el Problema 1 por BFS para el grafo3 es: 0.02214674

el tiempo para el Problema 1 por Disjoint-Set para el grafo3 es: 7.74140041

el tiempo para el Problema 1 por BFS para el grafo4 es: 0.004750890000000001

el tiempo para el Problema 1 por Disjoint-Set para el grafo4 es: 0.81701953

el tiempo para el Problema 1 por BFS para el grafo5 es: 0.00264042

el tiempo para el Problema 1 por Disjoint-Set para el grafo5 es: 0.06020834

el tiempo para el Problema 1 por BFS para el grafo6 es: 0.00329252

el tiempo para el Problema 1 por Disjoint-Set para el grafo6 es: 0.21177979

el tiempo para el Problema 1 por BFS para el grafo7 es: 0.00526267

el tiempo para el Problema 1 por Disjoint-Set para el grafo7 es: 4.70587641

el tiempo para el Problema 1 por BFS para el grafo8 es: 9.6013E-4

el tiempo para el Problema 1 por Disjoint-Set para el grafo8 es: 8.9222E-4

el tiempo para el Problema 1 por BFS para el grafo9 es: 0.00392657

el tiempo para el Problema 1 por Disjoint-Set para el grafo9 es: 0.02742304

el tiempo para el Problema 1 por BFS para el grafo10 es: 0.00227812

el tiempo para el Problema 1 por Disjoint-Set para el grafo10 es: 0.00616395

el tiempo para el Problema 2 por Lista Ordenada con Disjoint-Set SIN Heuristica para el grafo Conexo grafo1C es: 0.29352793

el tiempo para el Problema 2 por Lista Ordenada con Disjoint-Set CON Heuristica para el grafo Conexo grafo1C es: 1.0555655800000001

el tiempo para el Problema 2 por Heap con Disjoint-Set SIN Heuristica para el grafo Conexo grafo1C es: 1.13264443

el tiempo para el Problema 2 por Heap con Disjoint-Set CON Heuristica para el grafo Conexo grafo1C es: 0.94374055

el tiempo para el Problema 2 por Lista Ordenada con Disjoint-Set SIN Heuristica para el grafo Conexo grafo2C es: 3.63432809

el tiempo para el Problema 2 por Lista Ordenada con Disjoint-Set CON Heuristica para el grafo Conexo grafo2C es: 263.18701853

el tiempo para el Problema 2 por Heap con Disjoint-Set SIN Heuristica para el grafo Conexo grafo2C es: 47.026133560000005

el tiempo para el Problema 2 por Heap con Disjoint-Set CON Heuristica para el grafo Conexo grafo2C es: 42.14171551

el tiempo para el Problema 2 por Lista Ordenada con Disjoint-Set SIN Heuristica para el grafo Conexo grafo3C es: 0.006490020000000001

el tiempo para el Problema 2 por Lista Ordenada con Disjoint-Set CON Heuristica para el grafo Conexo grafo3C es: 0.019764459999999998

el tiempo para el Problema 2 por Heap con Disjoint-Set SIN Heuristica para el grafo Conexo grafo3C es: 0.010185729999999999

el tiempo para el Problema 2 por Heap con Disjoint-Set CON Heuristica para el grafo Conexo grafo3C es: 0.00920743

el tiempo para el Problema 2 por Lista Ordenada con Disjoint-Set SIN Heuristica para el grafo Conexo grafo4C es: 0.10740482000000001

el tiempo para el Problema 2 por Lista Ordenada con Disjoint-Set CON Heuristica para el grafo Conexo grafo4C es: 0.75052502

el tiempo para el Problema 2 por Heap con Disjoint-Set SIN Heuristica para el grafo Conexo grafo4C es: 0.33562034999999996

el tiempo para el Problema 2 por Heap con Disjoint-Set CON Heuristica para el grafo Conexo grafo4C es: 0.26072006

el tiempo para el Problema 2 por Lista Ordenada con Disjoint-Set SIN Heuristica para el grafo Conexo grafo5C es: 0.022961930000000002

el tiempo para el Problema 2 por Lista Ordenada con Disjoint-Set CON Heuristica para el grafo Conexo grafo5C es: 0.20576977999999999

el tiempo para el Problema 2 por Heap con Disjoint-Set SIN Heuristica para el grafo Conexo grafo5C es: 0.5647189399999999

el tiempo para el Problema 2 por Heap con Disjoint-Set CON Heuristica para el grafo Conexo grafo5C es: 0.29165744

el tiempo para el Problema 2 por Lista Ordenada con Disjoint-Set SIN Heuristica para el grafo Conexo grafo6C es: 0.10183876

el tiempo para el Problema 2 por Lista Ordenada con Disjoint-Set CON Heuristica para el grafo Conexo grafo6C es: 10.0361775

el tiempo para el Problema 2 por Heap con Disjoint-Set SIN Heuristica para el grafo Conexo grafo6C es: 1.7521323899999999

el tiempo para el Problema 2 por Heap con Disjoint-Set CON Heuristica para el grafo Conexo grafo6C es: 1.79888499

el tiempo para el Problema 2 por Lista Ordenada con Disjoint-Set SIN Heuristica para el grafo Conexo grafo7C es: 7.1105E-4

el tiempo para el Problema 2 por Lista Ordenada con Disjoint-Set CON Heuristica para el grafo Conexo grafo7C es: 0.00100544

el tiempo para el Problema 2 por Heap con Disjoint-Set SIN Heuristica para el grafo Conexo grafo7C es: 0.00198827

el tiempo para el Problema 2 por Heap con Disjoint-Set CON Heuristica para el grafo Conexo grafo7C es: 0.00173916

el tiempo para el Problema 2 por Lista Ordenada con Disjoint-Set SIN Heuristica para el grafo Conexo grafo8C es: 0.12296188000000001

el tiempo para el Problema 2 por Lista Ordenada con Disjoint-Set CON Heuristica para el grafo Conexo grafo8C es: 13.14927847

el tiempo para el Problema 2 por Heap con Disjoint-Set SIN Heuristica para el grafo Conexo grafo8C es: 1.9083551200000002

el tiempo para el Problema 2 por Heap con Disjoint-Set CON Heuristica para el grafo Conexo grafo8C es: 2.0655968799999997

el tiempo para el Problema 2 por Lista Ordenada con Disjoint-Set SIN Heuristica para el grafo Conexo grafo9C es: 0.00186594

el tiempo para el Problema 2 por Lista Ordenada con Disjoint-Set CON Heuristica para el grafo Conexo grafo9C es: 0.00718295

el tiempo para el Problema 2 por Heap con Disjoint-Set SIN Heuristica para el grafo Conexo grafo9C es: 0.003519

el tiempo para el Problema 2 por Heap con Disjoint-Set CON Heuristica para el grafo Conexo grafo9C es: 0.00406253

el tiempo para el Problema 2 por Lista Ordenada con Disjoint-Set SIN Heuristica para el grafo Conexo grafo10C es: 0.37971908000000004

el tiempo para el Problema 2 por Lista Ordenada con Disjoint-Set CON Heuristica para el grafo Conexo grafo10C es: 66.50307221

el tiempo para el Problema 2 por Heap con Disjoint-Set SIN Heuristica para el grafo Conexo grafo10C es: 9.121110029999999

el tiempo para el Problema 2 por Heap con Disjoint-Set CON Heuristica para el grafo Conexo grafo10C es: 8.99668528

Tabla de resultados empíricos para el ejercicio 1:

Ordenados primero de menor a mayor número de nodos y en segunda instancia por número de arcos de menor a mayor, tomados en Milisegundos

|  |  |  |  |
| --- | --- | --- | --- |
| Grafo | | BFS | Disjoint-Set |
| N | A |  |  |
| 2 | 1 | 0,016758 | 0,012681 |
| 5 | 10 | 0,179801 | 0,038043 |
| 50 | 49 | 0,044837 | 0,017211 |
| 71 | 900 | 0,211957 | 0,302989 |
| 179 | 179 | 0,139945 | 0,057971 |
| 190 | 300 | 0,155797 | 0,175724 |
| 200 | 15000 | 0,852355 | 2,921195 |
| 420 | 69870 | 1,264492 | 5,31139 |
| 500 | 40000 | 17,28396 | 44,821085 |
| 500 | 124750 | 2,03306 | 44,667553 |

Tabla de resultados empíricos para el ejercicio 2:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Grafo | | Ordenado (NanoSeg) | | Heap | |
| N | A | C/heurística | S/heurística | C/heurística | S/heurística |
| 2 | 1 | 0,024457 | 0,026721 | 0,024003 | 0,017663 |
| 10 | 10 | 0,038496 | 0,048007 | 0,025815 | 0,025816 |
| 20 | 30 | 0,100544 | 0,129528 | 0,136322 | 0,161231 |
| 80 | 100 | 4,818838 | 7,499997 | 3,368205 | 32,823355 |
| 124 | 6999 | 8,802533 | 9,802079 | 3,76902 | 5,442479 |
| 173 | 10000 | 13,22282 | 14,184776 | 2,157607 | 2,3048 |
| 211 | 210 | 0,83288 | 0,908061 | 1,253623 | 1,019021 |
| 300 | 41258 | 53,396715 | 65,639916 | 8,025359 | 8,637224 |
| 361 | 500 | 1,823369 | 2,287137 | 2,521285 | 5,153984 |
| 500 | 124750 | 338,578198 | 796,829361 | 83,122698 | 112,777576 |

SECCIÓN 3: CONCLUSIONES

Observando las tablas de resultados empíricos, es posible observar que:

* Para el Problema 1:
  + Si A >> N, el BFS tendrá un tiempo de ejecución menor al Disjoint-Set ya que el tiempo del disjoint set es dependiente de la cantidad de arcos presentes en el grafo mientras que el BFS NO necesita recorrer todos los arcos del grafo para determinar si un grafo es conexo o no
  + Si el grafo es ralo (esto es, A se acerca a N-1), el tiempo del disjoint set será mucho menor que el tiempo del BFS
  + Si A = N, el DIsjoint Set parece ser más eficiente que el BFS, esto se debe a que el grafo en si será ralo y por la observación anterior, el disjoint set será más eficiente en tiempo
* Para el Problema 2:

SECCIÓN 4: CODIGOS FUENTE

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