

PRÁCTICA 4

Ángela López López UO270318

EL ALGORITMO DEL VIAJANTE:

Se trata de un problema de **complejidad NP-completo**. En la teoría de la complejidad computacional esto quiere decir que para encontrar una solución a este problema se debe ejercer un esfuerzo computacional desmesurado e inadmisible hoy en día, ya que este crece de manera exponencial con la entrada del problema.

No existe una solución polinómica para este tipo de problemas.

Este algoritmo trata de responder a la pregunta (partiendo de una lista de ciudades y la distancia entre cada par de ellas):

¿Cuál es la ruta más corta posible que visita cada ciudad una única vez y al final regresa al punto de partida?

A pesar de que es uno de los problemas de optimización más estudiados (surgió en 1930 y se ha usado para numerosas investigaciones desde entonces) aún no se ha obtenido una solución adecuada, si bien es cierto que existe una gran cantidad de **heurísticos** capaces de simplificar la tarea.

Para ello se puede tratar este algoritmo como un **problema de grafos**.

Para ello se modela como si fuese un grafo no dirigido donde sus nodos son las ciudades y las aristas las distancias que las separan. De esta manera se puede buscar un **ciclo Hamiltoniano** (ya que sabemos que el grafo es completo y por lo tanto existe un ciclo hamiltoniano).

Como buscamos la ruta más corta posible, el ciclo Hamiltoniano de menor coste es la solución que necesitamos.

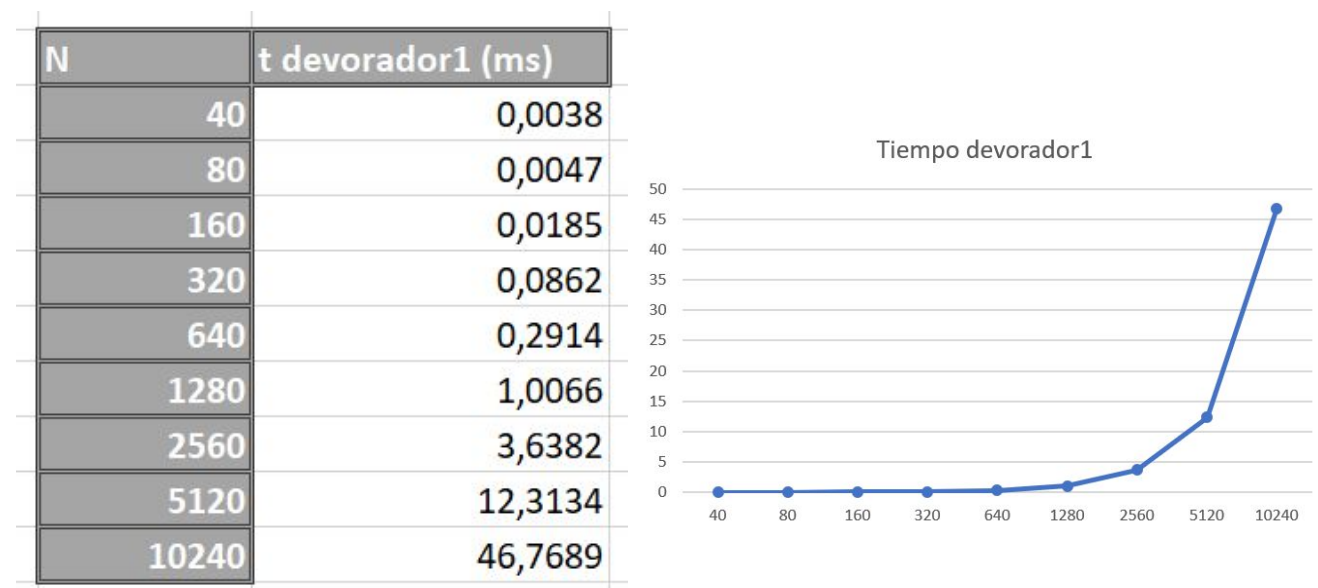
Para encontrarlo vamos a utilizar dos heurísticas distintas:

- **BUSCAR EL NODO MÁS CERCANO:** Para ello tenemos que calcular el peso de cada arista que parte del nodo actual al resto de nodos que no forman parte del camino ya recorrido. Para ello es necesario generar todas las aristas posibles desde el nodo inicial y comparar su coste. Seleccionamos la de menor coste y establecemos el nuevo nodo como pivote, desde el que se repite la acción. De esta forma al final tendremos el coste mínimo sumando los costes de cada arista. También se conoce como **camino de coste mínimo**, y tiene una complejidad $O(n*m)$, lo que en nuestro caso deriva en una complejidad cuadrática $O(n^2)$.
- **BUSCAR LA ARISTA MÁS CORTA:** Se escoge la arista más corta por la que aún no se ha recorrido, siempre que:

- No se forme un ciclo con su adición (excepto si todos los nodos ya han sido visitados).
- No sea la tercera arista en incidir en uno de los nodos seleccionados.

TIEMPOS DE LOS HEURÍSTICOS:

Como ya hemos mencionado anteriormente, el algoritmo del viajante tiene una complejidad exponencial que nos imposibilita tener resultados para un número muy grande de elementos en tiempos medianamente racionales. Es un problema muy complejo que se puede intentar abarcar con distintos heurísticos. Vamos a analizar el primero de ellos que se implementa en el método **devorador 1**, partiendo del primer nodo.



Como se puede apreciar tanto en la gráfica como en la tabla, la ejecución se corresponde con la complejidad teórica esperada $O(n^2)$.

A partir del número de elementos mostrados se para la ejecución pues la pila de la máquina virtual de java se desborda.