# BRANCH & BOUND

**ANÁLISIS Y DISEÑO DE ALGORITMOS.**

**Práctica Final**

**Memoria**

**Pedro Giménez Aldeguer, 15419933C**

**INDICE**

1. Estructuras de datos

1.1 Nodo

1.2 Lista de nodos vivos

2. Mecanismos de poda

2.1. Poda de tuplas no factibles

2.2. Poda de tuplas no prometedoras

3. Cotas pesimistas y optimistas

3.1 Cota pesimista y optimista del nodo inicial

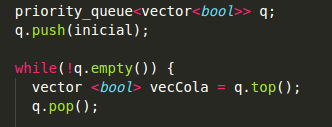
3.2 Cota pesimista y optimista de los demás nodos

4. Otros medios empleados para acelerar la búsqueda

5. Estudio comparativos de distintas estrategias de búsqueda

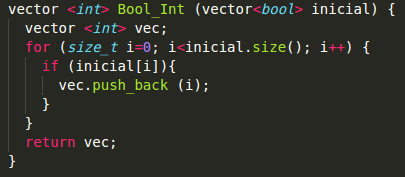
6. Soluciones y tiempos de ejecución.

**1. Estructuras de datos**

****

**1.1 Nodo**

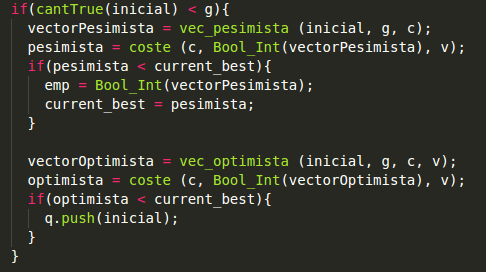
He utilizado un vector de tipo booleano como nodo, indicando true, las ciudades que tienen gasolinera y false, las que no. La posición en la que se encuentra el valor corresponde con la ciudad a la que hace referencia. Además, para sacar el coste, he tenido que utilizar un método que me transforma el vector de booleanos a un vector de enteros, ya que mi método “coste” trabaja con vector de enteros.



**1.2 Lista de nodos vivos**

Uso una cola de prioridad donde se van almacenando los nodos vivos, aquellos con posibilidades de ser expandidos, pero no he podido utilizar esta propiedad ya que no he sabido como hacerlo a pesar de haberme leído la teoría y haber ido a clase.

**2. Mecanismos de poda**

****

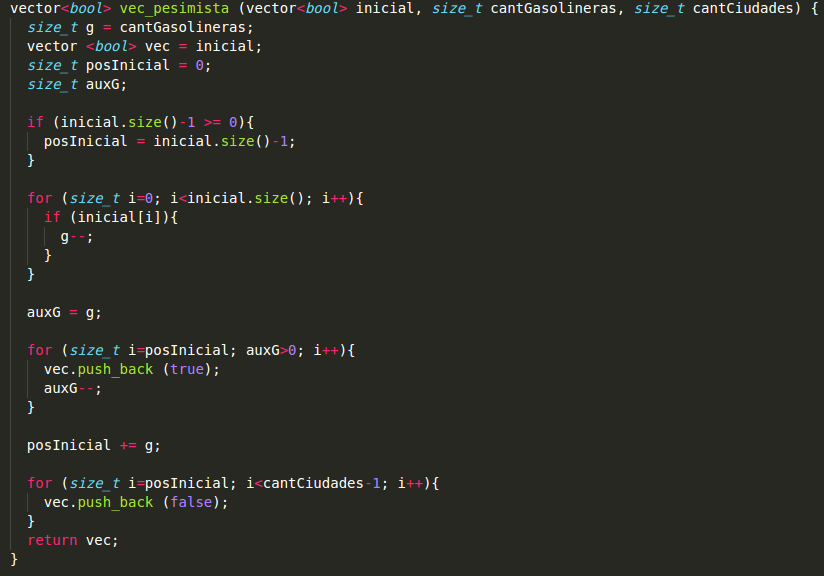
**2.1 Poda de tuplas no factibles**

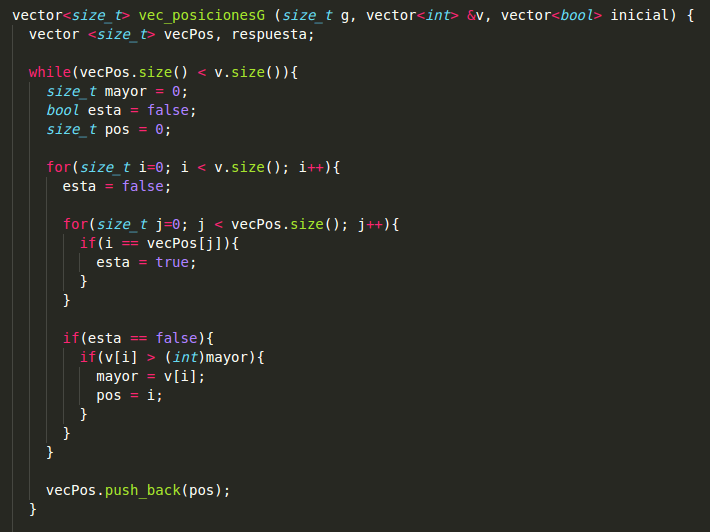
Los nodos no factibles los descarto al no crearlos, no permito crear nodos con mayores gasolineras de las que debemos colocar. Este tipo de poda es muy necesaria, es inútil e ineficiente calcular estos costes, ya que no podrían ser una solución factible.

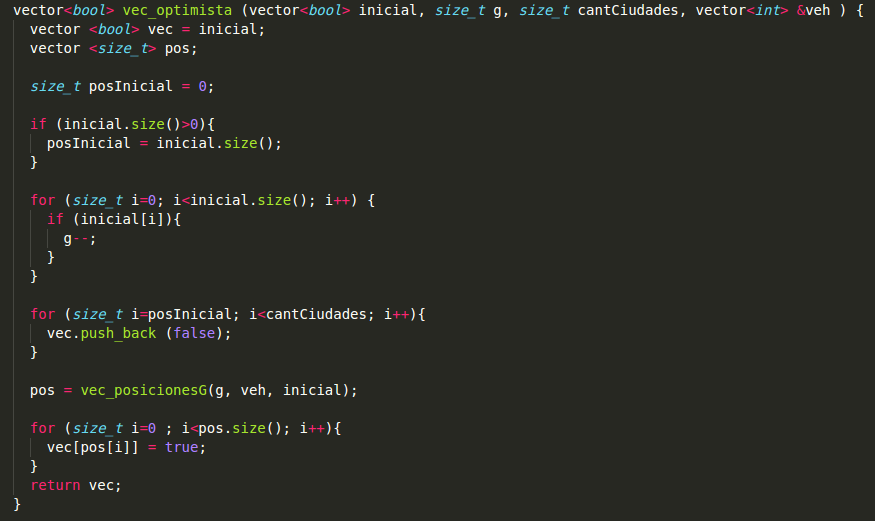
**2.2 Poda de tuplas no prometedoras**

No he sabido extraer el nodo más prometedor. Trata de no seguir por las ramas de los nodos que sabes que van a tener peor coste, se pasa por los nodos en los que vas a conseguir mejores resultados.

**3. Cotas pesimistas y optimistas**

****

****



**3.1 Cota pesimista y optimista del nodo inicial**

La cota pesimista del nodo inicial, calculando el coste a partir del vector pesimista, rellenado a true las n primeras posiciones.

La cota optimista del nodo inicial, calculando el coste a partir del vector optimista, rellenado a true las n posiciones de cuyas ciudades tengan más vehículos.

(n como el número de gasolineras)

**3.2 Cota pesimista y optimista de los demás nodos**

He utilizado el mismo método anterior para los demás nodos, pero teniendo en cuenta que no debo modificar las posiciones de los vectores que he extraído de la cola y los tomo como base.

El vector se rellena con todas las posiciones que faltan hasta que el tamaño de este sea igual a la cantidad de ciudades con el mismo método explicado anteriormente. Hay que tener en cuenta que la cantidad de gasolineras que debo de poner son n menos las que ya haya en el vector que tomo como base.

(n como el número de gasolineras)

**4. Otros medios empleados para acelerar la búsqueda**

No he empleado ningún otro mecanismo para acelerar la búsqueda.

**NO IMPLEMENTADO**

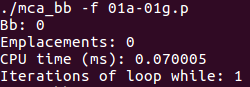
**5. Estudio comparativos de distintas estrategias de búsqueda**

No he podido hacer el estudio comparativo.

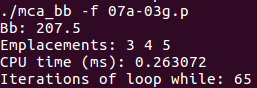
**NO IMPLEMENTADO**

**6. Soluciones y tiempos de ejecución.**

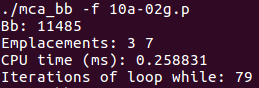
**Fichero 01a-01g.p:** Solución: 0; Tiempo de proceso: 0.070005 ms



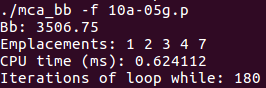
**Fichero 07a-03g.p:** Solución: 207.5; Tiempo de proceso: 0.263072 ms



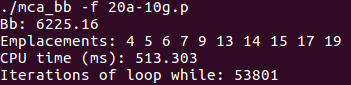
**Fichero 10a-02g.p:** Solución: 11485; Tiempo de proceso: 0.258831 ms



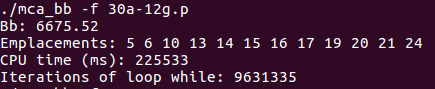
**Fichero 10a-05g.p:** Solución: 3506.75; Tiempo de proceso: 0.624112 ms



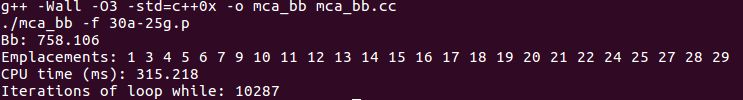
**Fichero 20a-10g.p:** Solución: 6225.16; Tiempo de proceso: 513.303 ms.



**Fichero 30a-12g.p:** Solución: 6675.52; Tiempo de proceso: 225533 ms.



**Fichero 30a-25g.p:** Solución: 758.106; Tiempo de proceso: 315.218 ms.



**Fichero 40a-10g.p:** Solución: ¿?



**Fichero 40a-30g.p:** Solución: ¿?

