Descripción del problema y método de resolución:

El problema nos pedía un número de pilotos k óptimo para un conjunto de trayectos dado.

Para ello debemos crear un grafo de demandas y cotas basado en los trayectos que se nos da como entrada, por cada trayecto creamos dos nodos, una vez creados todos los nodos debemos crear dos nodo s, t, s apunta a todos los origenes y todos los destinos apuntan a t.

Posteriormente con el grafo de demandas generado se transforma a un grafo de flujos mediante unas transformaciones, el grafo queda modificado de manera que en los nodos hay ciertas demandas y en sus aristas ya no hay cotas.

Generado el grafo, aplicamos el algoritmo de Edmons-Karp para encontrar el número de pilotos k que necessitamos.

Estructuras de datos usadas:

Hemos guardado todos los trayectos en un vector de estructuras, una estructura denominada trayecto que se define como:

```
struct trayecto{
	int IDor;
	int IDdest;
	int hsal;
	int hlleg;
};
```

Para poder trabajar sobre él a la hora de crear el grafo.

El grafo de demandas con cotas ha sido implementado con 2 estructuras de datos distintas, la primera una matriz de adyacencias, donde los datos que habia dentro de la matriz simbolizaban los arcos que habia entre cada nodo, y los arcos estaban definidos por una estructura que se componía de dos enteros, uno para las cotas (lower bound) y otro para sus respectivas capacidades.

Después de crear el grafo de demandas, pasábamos a transformarlo en un grafo de flujos. Para el grafo de flujos se ha utilizado también una matriz, en este caso de enteros, para representar los arcos que habia entre nodos. También hemos utilizado un vector de enteros para representar las demandas que habia en cada nodo.

Hemos considerado que una matriz de adyacencias podía ser más eficiente que una lista de adyacencias ya que en el caso que de los nodos salieran muchas aristas el coste temporal del programa implementado con listas de adyacencia se podria ver afectado.

Coste del algoritmo:

El coste temporal del algoritmo de Edmonds-Karp es de **$O(VE^2)$** donde V es el número de vértices del grafo de flujos y E sus respectivas aristas. donde V = 2n, suponiendo n el número de trayectos de entrada.(más los dos nodos s y t que creamos nosotros).

El coste espacial del programa es de **$O(2n^2)$** ya que para cada trayecto creamos dos nodos origen y destino.

Referéncias bibliográficas:

Sobre circulaciones con demandas y cotas:

http://www.cs.illinois.edu/class/fa07/cs473ug/Lectures/lecture19.pdf

Sobre reducciones de problemas de grafos con demandas y cotas a problemas de flujos:

http://www.cs.cmu.edu/~ckingsf/bioinfo-lectures/flowext.pdf

Sobre la generación del grafo para el problema de Air-Shceduling:

http://www.cs.princeton.edu/~wayne/kleinberg-tardos/pdf/07NetworkFlowII.pdf

Sobre el algoritmo Edmons-Karp:

http://www.cs.princeton.edu/~wayne/kleinberg-tardos/pdf/07NetworkFlowI.pdf