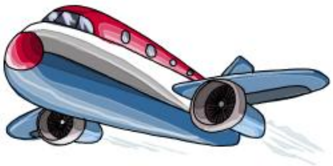
# Parte I Problema 1 - Plan de vuelo



## 1.1 Descripción del problema a resolver

En el presente problema, nos encontramos con una lista de vuelos internacionales entre distintos países. Cada vuelo tiene un origen, un destino y horarios de partida y llegada a cada uno de estos. Además, nos dan una ciudad A desde la cuál hay que partir y otra B a la cuál deseamos llegar. El objetivo es determinar un itinerario (es decir una combinación de los vuelos disponibles) que permita llegar de A a B lo más temprano posible, sin importar la cantidad de vuelos utilizada. Para hacer combinación entre dos vuelos, tienen que haber dos horas de diferencia entre los horarios de llegada y partida de ambos del aeropuerto. Veamos un ejemplo de instancia para el problema:

Ejemplo de instancia 1

```
bragado ezeiza 5
bragado rosario 3 5
rosario catamarca 7 9
bragado ezeiza 1 18
catamarca ezeiza 11 15
rosario ezeiza 7 100
```

Salida para instancia 1

```
15 3 1 2 4
```

Ejemplo de instancia 2

```
bragado ezeiza 5
bragado rosario 3 5
rosario catamarca 7 9
catamarca bragado 0 1
catamarca ezeiza 10 15
rosario ezeiza 7 12
```

Salida para instancia 2

```
12 2 1 5
```

En los ejemplos 1 y 2 se quiere encontrar un itinerario para viajar de bragado a ezeiza, elijiendo algunos de los 5 vuelos disponibles. En el primer ejemplo, se puede llegar como mínimo en el horario t = 15, tomando 3 vuelos. Y uno de los posibles itinerarios (podría haber más de uno que use 3 vuelos) sería tomar los vuelos 1, 2 y 4, en ese orden. Este itinerario de vuelos no podría realizarse en el ejemplo 2, porque el vuelo que une catamarca-ezeiza sale en t = 10, pero el único vuelo que llega a catamarca, lo hace en t = 9. Como hay menos de dos horas de diferencia entre que llega un vuelo y sale el otro, no es posible hacer esta combinación. Además, en el segundo ejemplo el mejor horario en el que se puede llegar a ezeiza es en t = 12, tomando los vuelos 1 y 5.

## 1.2 Ideas desarrolladas para la resolución

Para resolver este problema, consideramos a cada vuelo como un nodo. En base a las ciudades de origen y destino y horarios de salida y llegada de cada uno de los vuelos, utilizamos la técnica de programación dinámica para asignar a cada par de nodos una arista que los una si es que hay algún itinerario posible que permita pasar por ambos vuelos. Así, generamos un grafo dirigido en el cual para todo par de nodos A y B, hay una arista que los une si y sólo si se puede llegar desde el nodo A al nodo B, pasando por algún subconjunto del resto de los nodos. Representamos este grafo dirigido como una matriz A de $nxn$ donde n es la cantidad de nodos y donde $A_{ij}$ es 1 si hay un camino que une al nodo i con el nodo j y 0 si no. Esta matriz es llamada comúnmente la matriz de alcance de un digrafo.
Entonces, primero veamos qué es lo que hace nuestro algoritmo en los primeros 4 pasos, basándonos en el ejemplo 1:

$$\begin{vmatrix} 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{vmatrix} -> \begin{vmatrix} 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{vmatrix} -> \begin{vmatrix} 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{vmatrix} -> \begin{vmatrix} 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{vmatrix} -> \begin{vmatrix} 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{vmatrix}$$

Se puede ver que en la primera matriz, hay un 1 en la posición $(i, j)$ sii se puede hacer conexión directa entre los vuelos $i$ y $j$. En el siguiente paso, se actualiza la primera fila: desde el vuelo 1 puedo llegar a los vuelos que puedo llegar combinando de forma directa ó a los vuelos que puedo llegar combinando pasando antes por un vuelo intermedio. En el ejemplo, paso del vuelo 1 al vuelo 2 y luego, del vuelo 2 puedo pasar al vuelo 4. En los pasos siguientes la matriz permanece inalterada porque los vuelos 3, 4 y 5 van directo a ezeiza, y no hay ningún vuelo que salga de allí. Como ya analizamos para todo par de vuelos si era posible llegar de uno al otro, miramos la matriz que nos quedó. Mirándola con atención, se puede calcular, para cualquier par de ciudades, si es posible llegar de una a la otra y, si lo es, en que horario.
Veamos ahora con el ejemplo 2:

$$\begin{vmatrix} 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{vmatrix} -> \begin{vmatrix} 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{vmatrix} -> \begin{vmatrix} 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{vmatrix} -> \begin{vmatrix} 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{vmatrix} -> \begin{vmatrix} 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{vmatrix}$$

En este ejemplo, a diferencia de lo que sucede en el primero, en la posición $(2, 4)$ no hay un 1, pues no se puede combinar de forma directa a los vuelos rosario-catamarca con catamarca-ezeiza, por un tema de horarios de salida-llegada. En este caso, la posición $(1, 3)$ es igual a 1, porque se puede conectar de forma directa a los vuelos catamarca-bragado y bragado-rosario. En el primer paso no se puede agregar ninguna combinación para el vuelo 1, porque los vuelos 2 y 5 llegan ambos a ezeiza, y no hay ningún vuelo que salga de allí. En el siguiente paso, como el vuelo 2 llega a ezeiza, no se puede hacer nada por la misma razón. En el tercer paso, se puede añadir las combinaciones entre los vuelos 3-1 y 3-5, pues el vuelo 3 llega a bragado y de bragado salen dos vuelos: el 1 y el 5. Por la misma razón que no se pudo hacer nada para el vuelo 2, tampoco se puede hacer nada para los vuelos 4 y 5 y en los dos últimos pasos.

Entonces, como se ve en los ejemplos de más arriba, lo que hace nuestro algoritmo es ir vuelo por vuelo, actualizando para cada uno los vuelos a los cuáles puede ir y desde los cuáles se puede llegar. Al terminar de actualizar el último vuelo, en la fila $i$ de nuestra matriz va a formar un vector $v_i$, que en la posición $j$ va a tener un 1 si y sólo si hay un posible itinerario de vuelos que permite llegar desde el vuelo $i$ al vuelo $j$.

## 1.3 Justificación de cota de complejidad

Para poder cumplir con la complejidad requerida, que es de $O(n^2)$, representamos la matriz de la cuál hablamos en la sección anterior como un vector de enteros. De esta forma..

## 1.4 Testeos de complejidad