

## Segunda parte

Las autoridades de una ciudad deciden construir una red de subterráneos para resolver los constantes problemas de tráfico. La ciudad ya cuenta con  $N$  estaciones construidas, pero todavía no tienen ningún túnel que conecte ningún par de estaciones entre sí.

La red de subterráneos que se construya debe incluir a todas las estaciones (es decir, que de cualquier estación  $H$  pueda llegar a cualquier otra estación  $J$ , ya sea de manera directa o atravesando otras estaciones). Sin embargo, debido al acotado presupuesto, las autoridades desean construir **la menor cantidad de metros de túnel posibles**. Para esto han calculado cuantos metros de túnel serían necesarios para conectar de manera directa cada par de estaciones existentes.

### Objetivo

El objetivo de esta segunda parte del trabajo será resolver el problema planteado mediante dos técnicas algorítmicas distintas: Backtracking y Greedy.

Luego se deberán comparar los resultados teniendo en cuenta distintas métricas que permitan visualizar, mínimamente, la calidad de la solución y el costo de obtener dicha solución, con ambas técnicas.

### Implementación

La aplicación comenzará obteniendo la información de las estaciones y las distancias entre ellas de un archivo de texto, como el que se presenta a continuación.

Formato de archivo: Túneles.txt

<nombre\_estación\_1>;<nombre\_estación\_2>;<distancia>

Por ejemplo:

E1;E2;100

E2;E3;125

E3;E1;80

**Importante:** La ruta al archivo de texto de origen puede estar definida en el cuerpo del archivo main.cpp de la aplicación, solicitarse por consola o pasarse mediante argumentos al programa.

Una vez llevada a memoria la información de las estaciones y sus distancias, la aplicación deberá resolver el problema planteado mediante ambas técnicas. La solución deberá ser mostrada por consola presentando la siguiente información:

Técnica utilizada

Lista de túneles a construir (cada túnel se identifica mediante Estación 1-Estación 2)

Cantidad de metros totales a construir

Costo de encontrar la solución utilizando alguna métrica que permita medir este costo.

Por ejemplo:

```
Backtracking
E1-E2,E3-E1
180 kms
X metrica
```

## Recursos

Se deja disponible un proyecto base que contiene tres datasets propuestos para resolver el problema y un pequeño fragmento de código para poder abrir y leer, línea a línea, el contenido de un dataset indicado por parámetro.

## Informe

Se solicita la confección de un breve informe que responda tres cuestiones principales:

- ¿Cuál fue la estrategia Backtracking llevada adelante para resolver el problema? ¿Cuál es el costo computacional de dicha estrategia?
- ¿Cuál fue la estrategia Greedy llevada adelante para resolver el problema? ¿Cuál es el costo computacional de dicha estrategia?
- Mostrar una tabla comparativa, para las distintas entradas posibles, de los resultados obtenidos por ambas técnicas. Por resultados no sólo nos referimos a la calidad de la solución sino también a la métrica que determina el costo de obtener dicha solución.

Sumar al informe una portada y una breve conclusión del trabajo realizado.

## Requisitos de la entrega

Se deberá entregar un proyecto (junto al código fuente) que compile correctamente el código de la aplicación solicitada. También deberá entregarse un informe en versión digital editable que abarque los contenidos solicitados.

## Fecha de entrega y modalidad

La entrega final se realizará el **viernes 23 de junio**, siguiendo los mismos lineamientos que para la primera entrega. **Importante:** no se aceptarán entregas de la versión final fuera de término.