# Aplicación práctica de los Algoritmos Voraces (Kruskal y Prim)

## 1. "Asfaltado de la ciudad"

El Departamento de Tráfico de la ciudad ha decidido realizar trabajos de reparación por algunas calles. Pero para dar paso a estos trabajos es necesario cortar el tráfico por las calles en las que se estén llevando a cabo las reparaciones. Se solicita un algoritmo voraz que facilite la obtención de una franja *máxima* de calles en las que se puede interrumpir el tráfico, teniendo en cuenta que en todas las bifurcaciones que lleven a esas calles debería retenerse también el tráfico pero permitiendo/respetando el máximo flujo de tráfico posible *Atención*: ¿cuál es la función de selección?

Realiza una codificación que dé solución a este problema: (1) utilizando **Kruskal** (con la tercera aproximación de partición). (2) utilizando **Prim**.

Además de ambas codificaciones, se solicita (A) realizar un estudio analítico y (B) un estudio experimental, de las implementaciones realizadas:

- *Estudio analítico*: siguiendo el algoritmo propuesto, calcula las funciones de coste de ambas versiones, tanto para el **tiempo** consumido como para el **espacio extra** y expresa los resultados usando **notación asintótica**.
- *Estudio experimental o empírico*: teniendo en cuenta las dimensiones previamente obtenidas para cada implementación, realizar las mediciones empíricas de los tiempos de ejecución de cada implementación en los momentos críticos (*preparación* de los datos a partir del fichero de entrada, *ejecución* y *recogida de datos a partir* del fichero salida) para una colección significativa de datos de prueba, teniendo en cuenta diferentes tamaños. Las mediciones de experimentación deben presentarse en el informe utilizando tablas y gráficos, interpretando lo que reflejan y estableciendo coherencia con el resultado obtenido en el análisis analítico.

**Datos de entrada/salida y formatos para cada ejecución**

**Entrada**: los datos de entrada deben estar recogidos en un fichero de texto con el siguiente formato:

- La primera línea, contiene un número entero positivo, ***n***, ($n \leq 10^6$): corresponde al número de vértices que contiene el grafo.
- En la segunda línea, hay otro número entero positivo, ***a*** ($a \leq 10^8$): indica el número de aristas que contiene el grafo.
- Las siguientes ***a*** líneas contienen 3 números cada una:
    - Los dos primeros son números enteros, $x, y \in [0 \,..n\text{-}1]$; representan una arista del grafo
    - El tercero es un real positivo, $R^+$, que indica el peso correspondiente a la arista $(x, y)$.

**Salida**: los datos de salida deben guardarse en un fichero de texto con el siguiente formato:

- La primera línea debe contener el coste del árbol de expansión
- La segunda, un número natural, **k**, que indica la cantidad de aristas que contiene la solución.
- A continuación **k** líneas, cada una de las cuales contendrá 3 números:
    - Los dos primeros deben ser dos enteros positivos, *x y* pertenecientes al intervalo [0.. N-1], y representan la arista *(x, y)*;
    - El tercero, será un número real positivo, correspondiente al peso de la artista, (x, y).

## 2. "ARM que incluye el conjunto de aristas, T"

Dado un grafo conexo cuyas aristas tienen pesos positivos (llamemos, por ejemplo, GeL a dicho grafo), se pide construir un árbol de expansión pseudo-mínimo que contenga un subconjunto dado de aristas de GeL (llamemos, por ejemplo T a dicho subconjunto de aristas de GeL); es decir, construir el 'árbol' cuyo peso sea el menor de todos los que visitan todos los nodos de GeL y que contienen las arista indicadas T. **Nota**: todas las aristas de T deben estar en la solución, aunque cabe la posibilidad de que dicha solución no sea un árbol.

Partiendo del algoritmo de **Kruskal** (utilizando la tercera aproximación de partición) y de P**rim**, se pide desarrollar dos implementaciones diferentes para solucionar este ejercicio, eligiéndolas de entre las siguientes opciones:

a) Kruskal' (con 3ª aproximación de Partición);

b) Kruskal' (con 3º aproximación de Partición) + Montículo

c) Prim'

d) Prim' +Montículo

***Nota***: Kruskal' y Prim' hacen referencia a los algoritmos que debéis implementar adaptando **Kruskal** y **Prim** para que obtengan un *árbol de expansión pseudo-mínimo*.

Además de ambas codificaciones, se solicita (A) hacer un estudio analítico y (B) un estudio experimental, de las implementaciones realizadas:

- *Estudio analítico*: siguiendo el algoritmo propuesto, calcular las funciones de coste de ambas versiones, tanto para el **tiempo** consumido como para el **espacio extra** y expresar los resultados usando **notación asintótica**.
- *Estudio experimental* o empírico: teniendo en cuenta las dimensiones previamente obtenidas para cada implementación, realizar las mediciones de los tiempos de ejecución de cada implementación en los momentos críticos (*preparación de los datos* a partir del fichero de entrada, *ejecución* y *recogida de datos* en el fichero salida). Las mediciones de experimentación deben presentarse en el informe utilizando tablas y gráficos, interpretando lo que reflejan y estableciendo la coherencia con el resultado obtenido en el análisis analítico.

**Datos de entrada/salida y formatos para cada ejecución**

**Entrada**: los datos de entrada deben estar recogidos en un fichero de texto con el siguiente formato:

- La primera línea, contiene un número entero positivo, **n**, ($n \leq 10^6$): corresponde al número de vértices que contiene el grafo.
- En la segunda línea, hay otro número entero positivo, **a** ($a \leq 10^8$), que indica el número de aristas que contiene el grafo.
- Las siguientes **a** líneas contienen 3 números cada una:
    - Los dos primeros son números enteros, $x, y \in [0 ..n-1]$; representan una arista del grafo
    - El tercero es un número real positivo, R+, que denota el peso correspondiente a la arista (x, y).
- La siguiente línea contiene la cardinalidad del conjunto T, llamemos ***b*** a dicha cardinalidad.
- Las siguientes ***b*** líneas, representan las aristas del conjunto T, contiendo cada una de ellas, 3 números:
    - Los dos primeros, expresan la arista (x, y), siendo x e y valores en el intervalo[0..n-1]
    - El tercero es un número real positivo, R+, que denota el peso correspondiente a la arista (x, y).

**Salida**: los datos de salida deben guardarse en un fichero de texto con el siguiente formato:

- La primera línea debe contener el coste del árbol de expansión "pseudo mínimo".
- La segunda tendrá un número natural ***k***, que indica la cantidad de aristas que contiene la solución.
- En la tercera línea, debe guardarse un número entero positivo, ***Coste***, que indica el coste del árbol pseudo-mínimo de expansión.
- A continuación deben colocarse ***k*** líneas; cada una de las cuales tendrá 3 números:
    - los dos primeros deben ser dos enteros positivos, *x*, *y* pertenecientes al intervalo [0.. n-1], y representan la arista (*x, y*);
    - el tercero, debe ser un número real positivo $R^+$, que corresponde al peso de la artista, (*x, y*).