

## M23 Graphes - épisode 5

Un arbre couvrant minimal (ACM) d'un graphe est un arbre couvrant (sous-graphe acyclique connexe qui contient tous les sommets) dont la somme des poids des arêtes est minimale. Il n'existe pas d'autre arbre couvrant du graphe dont la somme des poids soit inférieure.

---

### 1 - Tri d'arêtes

Afin de mettre en oeuvre l'algorithme de Kruskal pour la construction d'un arbre couvrant minimal, on a besoin de trier les arêtes en fonction de leur poids. On commencera donc par écrire une fonction de tri d'un tableau d'arêtes.

*Dans le fichier `algos.c`, écrire la fonction suivante qui remplit le tableau `aretes_triees` (préalablement alloué) avec les arêtes du graphe `g` triées selon leur valuation croissante, valuations contenues dans le tableau `poids_arete` :*

```
void trier_aretes(arete *aretes_triees,  
                 graphe const *g, double const *poids_arete);
```

On pourra par exemple mettre en oeuvre l'algorithme de tri par insertion.

### 2 - Kruskal

On a maintenant tout ce qu'il nous faut pour écrire l'algorithme de Kruskal.

*Dans le fichier `algos.c`, écrire la fonction suivante qui génère le graphe `acm`, arbre couvrant minimal obtenu à partir du graphe `g` et des valuations d'arêtes contenues dans le tableau `poids_arete` en utilisant l'algorithme de Kruskal :*

```
void kruskal(graphe const *g, double const *poids_arete, graphe *acm);
```