M23 Graphes - épisode 4

Jusqu'à présent, on a représenté des graphes non-valués. Ce temps est révolu.

---

1 - Graphes valués

Dans notre structure de données, on représente les arêtes par des entrées dans un tableau contigu (le tableau `aretes`). Ainsi, chaque arête dispose d'un index entre `0` et `A-1` (`A` étant le nombre d'arêtes du graphe).

De la même manière qu'on a associé des valeurs aux sommets d'un graphe avec des tableaux dynamiques dont la taille est égale à l'ordre du graphe, on peut associer des valeurs aux arêtes d'un graphe en créant des tableaux dont la taille est égale au nombre d'arêtes du graphe.

Dans ces tableaux, la case d'index `i` contient les données associées à l'arête d'index `i` dans le graphe.

> ***! À nouveau, ne pas oublier de systématiquement libérer la mémoire allouée dynamiquement !***

---

2 - Initialisation

Afin de pouvoir tester nos algorithmes sur des graphes valués aux poids aléatoires, on se dote d'une nouvelle petite fonction utilitaire.

> ***Dans le fichier `utils.c`, écrire la fonction suivante qui initialise le tableau `val` de taille `n` (préalablement alloué) avec des flottants aléatoires dont la valeur est comprise entre 0 et `max` :***
>
> ```
> void init_aleatoire(double *val, size_t n, double max);
> ```

---

3 - Distance

> ***Dans le fichier `algos.c`, écrire la fonction suivante qui calcule et stocke la distance de chaque sommet du graphe `g` au sommet `s` passé en argument :***
>
> ```
> void dijkstra(graphe const *g, sommet s, double const *poids_arete,
>               double *distance_sommet);
> ```

Le tableau `poids_arete` reçu en argument contient les poids des arêtes de `g`. Le tableau `distance_sommet` (préalablement alloué) contiendra à l'issue de l'appel la distance au sommet `s` pour chaque sommet du graphe.

Comme le suggère le nom de la fonction, on utilisera l'algorithme de Dijkstra.

Afin d'initialiser les valeurs de distance, on pourra utiliser la constante `DBL_MAX`. Il faudra bien faire attention à ce que l'algorithme soit robuste dans le cas de graphes contenant plusieurs composantes connexes.