Devoir NF20 - Automne 2014

1. **Première partie** : algorithmes d'arbres couvrants.

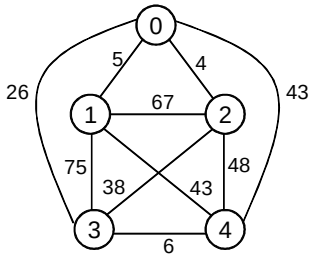
1.1 Vous devez implémenter les algorithmes de Prim, Kruskal et élimination de sous-cycle. Pensez aux structures de données pour stocker le graphe qui sera fourni en entrée de votre programme et pour stocker l'arbre couvrant.

Il existe plusieurs structures de données pour stocker des arbres. Un bon choix est déterminant dans la performance finale de l'algorithme. Par exemple, en utilisant un tas binaire minimal, les deux algorithmes peuvent avoir une complexité dans le pire des cas de O(|E| log |V|). L'algorithme de Prim peut être amélioré par l'utilisation d'un tas de Fibonnacci. De même, la structure Union-Find permet d'améliorer sensiblement les performances de l'algorithme de Kruskal dans les tests de fusion.

1.2 Les programmes réalisés doivent lire des instances de graphe au format suivant :

```
UNDIRECTED GRAPH
NB_NODES            5
NB_EDGES            10
LIST_OF_EDGES       COSTS
1       0       5
2       0       4
2       1       67
3       0       26
3       1       75
3       2       38
4       0       43
4       1       43
4       2       48
4       3       6
END
<Graph generated by Andréa Duhamel, 2012>
```

Cette instance correspond au graphe K_5 ci-dessous. Elle est disponible sur moodle.



2. **Deuxième partie** : analyse de complexité.

2.2 Analyse de complexité dans le pire des cas : vous allez évaluer de façon expérimentale la complexité des programmes réalisés. Un ensemble d'instances se trouve en ligne à cette fin. Vous dessinerez les courbes d'évolution du temps de calcul en fonction de la taille des instances et la courbe de la complexité théorique dans le pire des cas. Vous les inclurez dans le rapport final, accompagnées d'une analyse.

3. **Remise du travail**

3.1 Ce travail sera réalisé en binôme ou trinôme et rendu au plus tard le **06 janvier 2015 à midi.**

| Andréa Duhamel
Maître de conférences
ICD-LOSI, Université de Technologie de Troyes

3.2 Le rapport doit être imprimé et ne pas dépasser 5 pages (Format A4, police 12, simple interligne, marges à 2.0 cm). **Pour chaque page en trop, un point sera enlevé à la note finale.**

- le rapport doit contenir les analyses expérimentales de complexité des algorithmes développées. Signalez si vos codes arrivent à atteindre les complexités théoriques. S'il existe des écarts, expliquez où ils se trouvent. Fournissez le contexte expérimental :
  - machine de test
  - langage de programmation et version
  - choix des structures de données.

  Vos conclusions à propos de la complexité théorique et expérimentale sont également attendues.

3.3 Le code source, les exécutables et le rapport devront être placés dans une archive compactée à **vos noms** et déposée **sur moodle au plus tard le 06 janvier 2013 à midi** dans le répertoire de remise des projets. Je dois pouvoir compiler et exécuter votre source **sans erreurs**. Attention : le rapport doit être aussi être rendu dans une version **imprimée au plus tard le 06 janvier 2013 à midi.**

3.4 Vous allez présenter votre travail dans la dernière séance de TD et les programmes seront testés sur un groupe d'instances spéciales qui vous sera fourni à ce moment-là. Les détails du déroulement de cette étape seront donnés prochainement.

| Andréa Duhamel
Maître de conférences
ICD-LOSI, Université de Technologie de Troyes