
Algorithmique Avancée

Format de rendu du dernier TP

1 Code (sur 5 points)

Le code sera évalué automatiquement. Un script lancera vos programmes. Il vérifiera que les solutions que renvoient vos programmes sont bien des solutions et il évaluera ces solutions. Afin d'éviter les aléas dus au hasard, vos programmes seront lancés plusieurs fois sur le même jeu de tests et seule la moyenne des temps d'exécution et des solutions seront pris en compte.

Pour que le script puisse noter votre code, il est indispensable que vous respectiez les consignes suivantes.

- Vous ne devez déposer sur Moodle qu'une archive. Cette archive doit être au format tar.gz (extension `tar.gz` ou `tgz`), au format tar.bz2 (extension `tar.bz2` ou `tbz2`) ou au format zip (extension `zip`). Aucun autre format ne sera accepté (notamment RAR et 7z). L'extension du nom de l'archive doit correspondre à son type.
- L'archive doit contenir un dossier à votre nom. Tous vos fichiers (code et rapport) doivent se trouver dans ce dossier.
- Le dossier doit contenir un makefile. Ce makefile doit permettre de lancer les commandes `make` et `make clean` dans votre dossier sans que cela ne génère une erreur. La commande `make` doit s'assurer que les fichiers `run_nc` et `run_wc` se trouvent dans le dossier et sont exécutables. La commande `make clean` doit nettoyer les éventuels fichiers intermédiaires.
- Le dossier doit contenir un fichier nommé `INSTALL` décrivant l'environnement dans lequel doivent être lancés vos programmes. Il devra préciser notamment la liste des utilitaires nécessaires à la compilation et/ou à l'exécution de vos programmes ainsi que les versions de ces utilitaires que vous avez utilisées pour tester votre code, par exemple gcc 4.8.3, python 2.7.9, Java Oracle JDK 1.8.0.112 ou GNU bash 4.2.53. Ce fichier sera éventuellement lu par un correcteur si la commande `make` échoue.
- Après l'exécution de la commande `make`, votre dossier doit contenir les exécutables `run_nc` et `run_wc`. Le premier programme doit permettre de résoudre un problème sans tenir compte des contraintes (`nc` pour « no constraints »), alors que le second doit prendre en compte les contraintes (`wc` pour « with constraints »). Chacun de ces programmes sera lancé avec deux arguments : le premier indique un temps maximal de calcul en secondes et le second indique le nom du fichier contenant le problème. Chaque programme doit afficher dans le temps imparti sur la sortie standard (`stdout`) une solution au problème. Le temps d'exécution des programmes sera mesuré et comparé au temps maximum donné en argument.
- La solution doit être affichée au format dot, le dépôt et les villes à livrer étant représentés par leur identifiant. Il n'est pas nécessaire de placer les villes selon leurs coordonnées.

2 Rapport (sur 15 points)

Votre rapport doit se trouver dans le même dossier que votre code, sous la forme d'un fichier au format PDF d'au plus trois pages. Il devra contenir les informations suivantes :

- votre nom et votre prénom,
- le choix du langage de programmation,
- la représentation du problème et des solutions,
- le calcul des voisinages (pour le problème sans contraintes et avec contraintes),
- les choix des voisins,
- vos démarches d'amélioration et de réglage,
- les résultats obtenus et leur analyse.

La qualité de la présentation et la correction du langage seront pris en compte.