

Devoir n° 2

(à rendre par groupe de 4 au plus tard le lundi 08 janvier 2018 à 8h00)*

Ça y est ! votre équipe est finaliste du Google HashCode 2019 et déjà de nombreuses entreprises se disputent pour vous offrir un emploi avec un salaire mensuel à 5 chiffres. Votre secret ? Vous aviez fait le mini-projet 2 du module Algorithmique et Complexité qui vous a permis de comparer différentes heuristiques d'un problème NP-complet et cet expertise vous a permis facilement de passer la phase éliminatoire du concours. Mais maintenant retour à fin 2017 où vous devez faire ce fameux mini-projet.

Le problème à traiter est la comparaison de plusieurs algorithmes pour choisir le meilleur d'entre eux.

Vous avez un problème de bin-packing à résoudre pour un client industriel. Votre client dispose d'objets à transporter, que vous devez traiter au moment de leur arrivée. [L'extrait d'article](#) ci-joint contient une description des algorithmes d'approximation suivants : *Next Fit*, *First Fit*, *Best Fit*, *Worst Fit* et *Almost Worst Fit*.

Ainsi vous disposez de 5 algorithmes différents. Vous devez donc préparer un dossier d'analyse, donner des résultats des mesures de simulation que vous avez effectuées et donner vos conclusions et recommandations. Le dossier doit être tel que vous puissiez le présenter à un industriel, pour le convaincre de faire le bon choix (et d'acheter la solution que vous préconisez).

Langage au choix C/C++, Java ou Python3 (un seul langage par projet).

Vous avez à rendre par dépôt DropBox à

<https://www.dropbox.com/request/8EquGkhq83eeg9p3pz1g>

un fichier intitulé `ALGOCOMP2_NOM1_NOM2_NOM3_NOM4.zip` **contenant un unique dossier** contenant votre travail (dossier nommé `ALGOCOMP2_NOM1_NOM2_NOM3_NOM4`).

- Le dossier doit être lui-même contenir trois répertoires, nommés `source`, `doc` et `exemples` et un fichier `makefile` qui crée deux exécutables dans le répertoire principal, nommés `algo.exe` et `stat.exe`. Les exécutables peuvent être des scripts en cas de langage non compilé.
 - Le premier permet l'exécution des 5 algorithmes sur les exemples fournis ([exemple100.txt](#), [exemple500.txt](#), [exemple1000.txt](#)) et sur un exemple que vous donnez, dans un fichier ascii nommé "monexemple.txt".
 - Le second effectue une série de simulations sur des exemples générés de manière aléatoire (le nombre et la manière de générer seront fourni par une phase de dialogue input/output).
- Dans le répertoire source uniquement des fichiers `.cc` et `.h`, ou `.java` ou `.py` sans duplication du contenu. Eventuellement un fichier `README`.
- Dans la documentation jointe (répertoire `doc`) un ou plusieurs fichiers **pdf** décrivant votre travail, votre analyse et vos conclusions.

La note tiendra compte de la clarté de **la rédaction** et de **la présentation** des simulations.

- * Les devoirs rendus par groupe de 3 élèves ou moins seront **pénalisés**. 60 étudiants sont inscrits en SI4 à ce module, il doit donc y avoir exactement 15 groupes.