Algorithmique – Programmation Objet – Python – a.u. 2017/2018
Contrôle intermédiaire N° 1 – mardi 7 novembre 2017 (durée 1h30)

Licence en Sciences Fondamentales, 2ème année
Université de Nice-Sophia Antipolis

Toutes les réponses doivent être justifiées. Les algorithmes doivent être écrits en pseudo-langage, lisibles et bien alignés. Le correcteur attachera de l'importance à la qualité de rédaction. Les documents, calculatrices ou téléphones portables ne sont pas autorisés. Le barème, sur 20, est donné à titre indicatif.

1 Échauffement (6 points)

1. (4 points) Écrire un algorithme avec le plus faible coût en temps possible pour calculer la multiplication de deux entiers relatifs a et b *sans utiliser la multiplication.*
2. (2 points) Écrire un algorithme qui compte le nombre d'occurences d'une valeur x, de type entier, dans un tableau T non trié de taille n.

2 Tri d'une liste chaînée par sélection (6 points)

Sur un tableau de n éléments (numérotés de 1 à n), le principe du tri par sélection est le suivant :
- rechercher le plus petit élément du tableau, et l'échanger avec l'élément d'index 1 ;
- rechercher le second plus petit élément du tableau, et l'échanger avec l'élément d'index 2 ;
- continuer de cette façon jusqu'à ce que le tableau soit entièrement trié.

Suivant ce principe, mais en remplaçant le tableau par une liste chaînée :

1. (4 points) Écrivez le pseudo-code de l'algorithme de tri par sélection sur une liste simplément chaînée. [1]
2. (2 points) Donnez son coût dans le pire des cas et dans le meilleur. Comparez ce coût au coût de l'algorithme sur un tableau donné ci-dessus.

3 Application d'une fonction aux sous-ensembles d'une liste (8 points)

Soit une liste L doublement chainée [2] énumérant un ensemble d'éléments $[x_1, \ldots, x_n]$. On possède une fonction F prenant une liste en argument, que l'on souhaite appeler sur chacun des sous-ensembles de L. Comment effectuer cette opération sans faire de copie de L en mémoire ?

Suggestions :
- Il faudra modifier L entre des appels successifs à F.
- Contrairement à un tableau, une liste permet de supprimer et insérer des éléments facilement.
- Une solution basée sur la récursion (diviser pour régner) semble prometteuse.

---

1. On supposera que les attributs suivants sont définis (L représente la liste entière et x est un élément de la liste) :
— x.DONNÉE : contient une référence à la donnée associée à l'élément x.
— x.SUIVANT : contient une référence à l'élément suivant l'élément x.
— L.PREMIER : contient une référence au premier élément de la liste, NIL si la liste est vide.

2. On utilisera les mêmes conventions que dans l'exercice 2 ; en outre, x.PRÉCÉDENT contiendra une référence à l'élément précédent de l'élément x.