

Algorithmique IV (UE-41) - TD 9.

TD 9. Tri par dénombrement, par répartition et tri lexicographique¹

EXERCICE 1. (1) Calculez la dispersion $\Delta(L)$ de chacune des listes suivantes :

(a) $L := [1, 1, 2, 3, 1, 8, 1, 2, 1, 1]$,

(b) $L := [3, 1, 1, 2]$.

(2) Trouvez un encadrement des valeurs de la fonction de dispersion. La borne inférieure est-elle atteinte ? Même question avec la borne supérieure. Justifiez.

EXERCICE 2. Démontrez que l'ensemble des entiers relatifs \mathbb{Z} est équipotent à l'ensemble des entiers naturels \mathbb{N} . Indication : explicitez l'application $\iota : \mathbb{Z} \rightarrow \mathbb{N}$ qui associe les nombres négatifs aux entiers impairs et les nombres positifs aux entiers pairs et montrez qu'elle est bijective.

EXERCICE 3. L'algorithme de comparaison lexicographique est rappelé en ALGO. 1.

(1) Démontrez que l'algorithme s'arrête.

(2) Faites la preuve de correction partielle.

EXERCICE 4. (1) Modifiez l'algorithme [TriRépartition](#) pour écrire un algorithme `Partition(@L)` qui partitionne une liste L de mots en listes L_k des mots de longueur k et renvoie la table P de ces listes. À l'issue du partitionnement, la liste L est vide. On utilisera librement l'algorithme `SupTete(@L)` qui renvoie la valeur à la tête de la liste L et supprime cette cellule, et l'algorithme `InsQueue(@L, x)` qui insère une liste atomique d'une seule cellule contenant x à la queue de la liste L .

(2) Calculez la complexité de l'algorithme qui réalise cette partition. On supposera que l'algorithme `InsQueue(@L, x)` est en $\Theta(1)$.

EXERCICE 5. Dans le tri lexicographique, pourquoi avant le tri par répartition sur la k -ème lettre, faut-il concaténer la liste L_k des mots de longueur k avant la liste L des autres mots ?

```
.....  
ALGORITHME COMPARAISON-LEXICO(u,v) : {-1,0,+1}
```

```
DONNEES
```

```
· u,v: mots sur un alphabet
```

```
VARIABLES
```

```
· i: entier
```

```
DEBUT
```

```
· i ← 1
```

```
(a) · TQ ((i ≤ |u|) ET (i ≤ |v|) ET (u[i] = v[i])) FAIRE
```

```
· · i ← i + 1
```

```
· FTQ
```

```
(b) · SI ((i > |u|) ET (i > |v|)) ALORS
```

```
· · RENVoyer 0
```

```
(c) · SINON SI ((i > |u|) OU ((i ≤ |v|) ET (u[i] < v[i]))) ALORS
```

```
· · RENVoyer -1
```

```
(d) · SINON
```

```
· · RENVoyer +1
```

```
· FSI
```

```
FIN
```

```
.....  
ALGO. 1. Comparaison pour l'ordre lexicographique.
```

1. Version du 13 janvier 2025, 11 : 22