



Algorithmique des tableaux (C5-160211-INFO)

Licence 1 - Année 2019/2020

TP 2 - Plages et tri

C. Saint-Jean, L. Mascarilla, E. Zahzah



Exercice 1: Plages

- Écrire une fonction *longss1* qui prend en entrée une séquence (Tableau ou chaîne de caractères) et un élément x du même type. Celle-ci doit retourner un tuple contenant la position de la plus longue sous-séquence consécutive contenant x ainsi que sa longueur. Par exemple :

"aabbcccccaaca", "b" \rightarrow (2, 3)

- On étend maintenant où seule la séquence est donnée. Vous devez déterminer en une seule passe quel est l'élément présent dans la plus longue sous-séquence consécutive. La fonction *longss2* retournera un triplet comme suit :

"aabbcccccaaca" \rightarrow ("c", 5, 4)

- On souhaite maintenant connaître la plus longue sous-séquence consécutive pour chacun des éléments de la séquence. Pour cela, la fonction *longss3* retournera un dictionnaire comme par exemple :

"aabbcccccaaca" \rightarrow {"a" : (9, 3), "b" : (3, 3), "c" : (5, 4)}

Exercice 2: Tri à Bulles

Voici l'algorithme du tri à bulles :

Algorithme 2 : Tri à bulles

Données : T un tableau de n éléments

Output : T est trié

```
1 pour fin  $\leftarrow$  (n - 1) à 1 faire
2   pour j  $\leftarrow$  1 à fin faire
3     si T[j + 1] < T[j] alors
4       échanger T[j + 1] et T[j];
```

- Implémentez cet algorithme en *Python* .
- Si aucun échange n'est effectué lors d'une la boucle interne sur 'j', on peut remarquer que le tableau est déjà trié. Modifier votre programme pour mettre en œuvre cette optimisation.
- De la même façon qu'en cours, mettez en place une évaluation de cet algorithme :
 - Vérifiez que le tableau est trié.
 - Temps pour le meilleur des cas, pire des cas pour 30000 éléments
 - Comptez les échanges et les comparaisons.