

1 Oups !

Question 1. Yorel Reivax a écrit le code suivant pour le tri fusion. Quelle est son erreur ?

```
let rec merge_sort l =  
  let a, b = split l in  
  merge (merge_sort a) (merge_sort b)
```

2 Tri rapide

L'algorithme de tri rapide consiste à choisir un pivot et à trier récursivement d'un côté les éléments plus petits que le pivot et de l'autre côté les éléments plus grands. Il tient son nom du fait que son implémentation sur des tableaux est très efficace en pratique.

Question 2. Quelle est la complexité du tri rapide ?

Question 3. Les performances du tri rapide dépendent beaucoup du choix du pivot, quelle semble selon vous la meilleure stratégie pour choisir le pivot ?

Question 4. Implémentez l'algorithme de tri rapide pour des tableaux¹. Le tri doit être fait en place (*i.e.* sans utiliser de tableau supplémentaire).

Question 5. Implémentez l'algorithme de tri rapide pour des listes.

3 Sélection

On s'intéresse au problème de sélection qui consiste à trouver le $k^{\text{ème}}$ plus petit élément d'une collection (tableau ou liste).

Question 6. Écrivez une fonction `selection` de type `int -> 'a list -> 'a` telle que `selection k l` renvoie le $k^{\text{ème}}$ plus petit élément de la liste `l`.

Question 7. Étant donnée une fonction `median` qui renvoie la médiane d'une liste, trouvez un algorithme de sélection en temps linéaire.

Question 8. Adaptez votre algorithme pour qu'il soit correct même quand la fonction `median` renvoie une médiane approximative (au moins 30% des éléments sont plus petits et au moins 30% sont plus grands).

Question 9. Montrez qu'en groupant les éléments par 5 et en prenant la médiane des médianes de chaque groupe de 5, on obtient bien une médiane approximative.

Question 10. * Écrivez une nouvelle fonction `selection` de complexité linéaire.

1. Dans tout ce sujet on utilisera la relation d'ordre correspondant à l'opérateur `<` de `caml`.

4 Tri rapide pire des cas

Question 11. Maintenant que vous savez résoudre le problème de sélection en temps linéaire, trouvez une stratégie pour choisir le pivot du tri rapide qui garantisse une complexité $O(n \log(n))$ dans le pire des cas.

Question 12. Implémentez cette stratégie pour les listes.

Question 13. * Implémentez cette stratégie pour les tableaux. On souhaite conserver le caractère « en place » du tri.