

TD n°2

1 Échauffement : compter en base d'interrupteurs

On considère une ligne de n interrupteurs tous en position fermée. On s'amuse ensuite à compter en base 2 avec ces interrupteurs, en partant de 0, jusqu'à $2^n - 1$. Calculez le nombre total de changements d'état des interrupteurs, au choix :

- en comptant le nombre de changements d'état quand on passe de m à $m + 1$,
- en sommant le nombre de changements d'état de chaque interrupteur,
- par de l'analyse amortie.

2 Plat principal : recherche de mots

Dans cette partie, on se donne un ensemble de vrais mots de la vraie vie, D . Par exemple "tas", "tige", "table", "tdman", "tabouret" ... L'alphabet que nous appellerons Σ est alors celui que vous connaissez depuis que vous êtes petits, avec toutes les lettres de 'a' à 'z'.

Question 1 Quel est le cardinal de Σ ?¹

□

On cherche un moyen de savoir si un mot m appartient à D , de préférence rapide. On veut aussi, si $m \notin D$, trouver u préfixe de m de taille maximale et tel qu'il existe $v \in \Sigma^*$ tel que $uv \in D$.

Question 2 Réécrire cette question en langage compréhensible et donner un exemple qui clarifie ce charabia. Pouvez-vous utiliser le TD n°1 pour résoudre ce problème?

□

Question 3 Est-ce que ça vous fait penser à un automate? Si oui, représentez cet automate sous forme d'arbre. Si non, retournez à la question 3.

□

Question 4 Pensez à une implantation d'une structure de données pour votre arbre.

□

¹Attention, il n'y a pas de piège.

Question 5 Vérifiez que vous n'arrivez pas à avoir une complexité en $O(|m|)$ pour la recherche d'un mot m . Souvenez-vous maintenant que vous n'avez pas de contrainte de taille, et donc vous pouvez utiliser toute la mémoire que vous voulez. Trouvez donc une meilleure structure qui vous permette d'atteindre cette complexité.

□

Question 6 Rendez-vous compte avec horreur du temps que prend l'initialisation de votre structure de donnée. Inventez un moyen qui vous permette de n'initialiser que les données qui vous intéressent.

□