

Activité sur les arbres binaires de recherche

On veut créer un dictionnaire un peu particulier qui permette de retrouver très rapidement un mot. On propose un fonctionnement un peu particulier : Plutôt que de stocker les mots dans une liste, les mots sont stockés les uns par rapport aux autres.

Pour cela, on va définir quelques règles :

- Les mots sont rangés par ordre alphabétique
- Chaque mot peut avoir jusqu'à deux mots associés : un mot plus petit et un mot plus grand
- Le mot associé à gauche est plus petit que le mot courant
- Le mot associé à droite est plus grand que le mot courant
- Tous les mots vont partir d'un mot initial

Voilà un schéma pour placer les mots :

graph TD

A[] --> B[]

A --> C[]

B --> E[]

B --> F[]

C --> G[]

C --> H[]

E --> I[]

E --> J[]

F --> K[]

F --> L[]

G --> M[]

G --> N[]

H --> O[]

H --> P[]

1. Placez les mots suivants dans le dictionnaire :

- maison
- lapin
- pomme
- arbre
- chat
- abeille
- navire
- oiseau

Maintenant pour s'assurer qu'il soit utilisable, on vérifie que les mots sont bien rangés par ordre alphabétique. En lisant les mots de gauche à droite, on doit avoir un ordre croissant. Sinon, c'est qu'il faut vérifier une dernière règle : Tous les mots à gauche d'un mot doivent être plus petits que lui et tous les mots à

droite d'un mot doivent être plus grands que lui. Si ce n'était pas déjà le cas, on peut réorganiser les mots pour que ce soit le cas.

Correction

```
graph TD
  A[Maison] --> B[Lapin]
  A --> C[Pomme]
  B --> E[Arbre]
  B --> F[ ]
  E --> I[Abeille]
  E --> J[Chat]
  C --> G[Navire]
  C --> H[ ]
  F --> M[ ]
  F --> N[ ]
  G --> O[ ]
  G --> P[Oiseau]
  H --> Q[ ]
  H --> R[ ]
```

2. Quelle règle faut-il suivre pour ajouter un mot dans le dictionnaire ? Ecrivez un algorithme en pseudo-code qui permette d'ajouter un mot dans le dictionnaire.

Correction

Pour ajouter un mot dans le dictionnaire, il faut :

- On part du mot initial
- Si le mot est plus petit que le mot courant, on va à gauche
- Sinon si le mot est plus grand que le mot courant, on va à droite
- Si il y a déjà un mot à gauche ou à droite, on recommence à partir de ce mot
- Sinon on ajoute le mot au nouvel emplacement

3. On peut maintenant chercher un mot dans le dictionnaire. Ecrivez un algorithme en pseudo-code qui permette de chercher un mot dans le dictionnaire.

Une fois que vous avez écrit l'algorithme, donnez votre algorithme à un camarade et demandez lui de chercher un mot dans le dictionnaire. Il choisi un mot et tente de le retrouver grâce à votre algorithme. A chaque étape, il détaille ce qu'il fait jusqu'à ce qu'il trouve le mot ou qu'il ne le trouve pas.

Si il n'arrive pas à trouver le mot, vous pouvez réfléchir ensemble à comment corriger l'algorithme.

Correction

Pour chercher un mot dans le dictionnaire, il faut :

- On part du mot initial
- Si le mot est égal au mot courant, on a trouvé le mot
- Sinon si le mot est plus petit que le mot courant, on va à gauche

- Sinon si le mot est plus grand que le mot courant, on va à droite
- Si on arrive à un mot vide, on a pas trouvé le mot
- Sinon on recommence à partir de ce mot

4. Un mot peut-il être présent à plusieurs endroits dans le dictionnaire ?

Correction

Non, un mot ne peut être présent qu'à un seul endroit dans le dictionnaire. En effet, si on ajoute un mot qui est déjà présent, on va le chercher dans le dictionnaire et on va le retrouver. On va donc le remplacer par le nouveau mot.

5. Peut-on utiliser ce dictionnaire avec autre chose que des mots ? Qu'est ce qui est nécessaire pour pouvoir utiliser ce dictionnaire ?

Correction

Oui, on peut utiliser ce dictionnaire avec autre chose que des mots. Par exemple, on peut utiliser ce dictionnaire pour ranger des nombres. On peut aussi utiliser ce dictionnaire pour ranger des objets tant qu'on peut les comparer entre eux.

6. Au vu de la forme du dictionnaire, au maximum combien de mots peut-on ranger dans le dictionnaire en fonction de la hauteur du dictionnaire ? (On appelle hauteur du dictionnaire le nombre d'étages du dictionnaire).

Correction

On peut ranger au maximum 2^{hauteur} mots dans le dictionnaire. En effet, à chaque étage, on peut avoir deux mots. On peut donc avoir 2^{hauteur} mots au maximum.

7. Comparons avec un dictionnaire classique. En supposant qu'on lise les mots un par un, les uns à la suite des autres dans un dictionnaire classique, combien d'étapes faut-il faire pour retrouver un mot dans le dictionnaire ? (On l'exprimera en fonction du nombre de mots dans le dictionnaire).

Correction

Pour retrouver un mot dans un dictionnaire classique, il faut lire tous les mots jusqu'à trouver le mot. On peut donc retrouver un mot en nombre de mots étapes.

8. Avec ce dictionnaire, combien d'étapes faut-il faire pour retrouver un mot dans le dictionnaire ? (On l'exprimera en fonction du nombre de mots dans le dictionnaire). Comparer avec la question précédente, que peut-on en conclure ?

Correction

Au maximum, on peut retrouver un mot en hauteur étapes. En effet, à chaque étape, on peut éliminer la moitié des mots. On peut donc retrouver un mot en hauteur étapes. On a dis précédemment que le nombre de mots maximum était 2^{hauteur} . On peut donc retrouver un mot en \log_2 nombre de mots étapes.

```
graph TD
  A[10] --> B[5]
```

```
A --> C[_____]
B --> E[2]
B --> F[_____]
```

Ce dictionnaire est donc beaucoup plus efficace que le dictionnaire classique car le nombre d'étapes pour retrouver un mot est beaucoup plus faible (\log_2 nombre de mots étapes contre nombre de mots étapes).