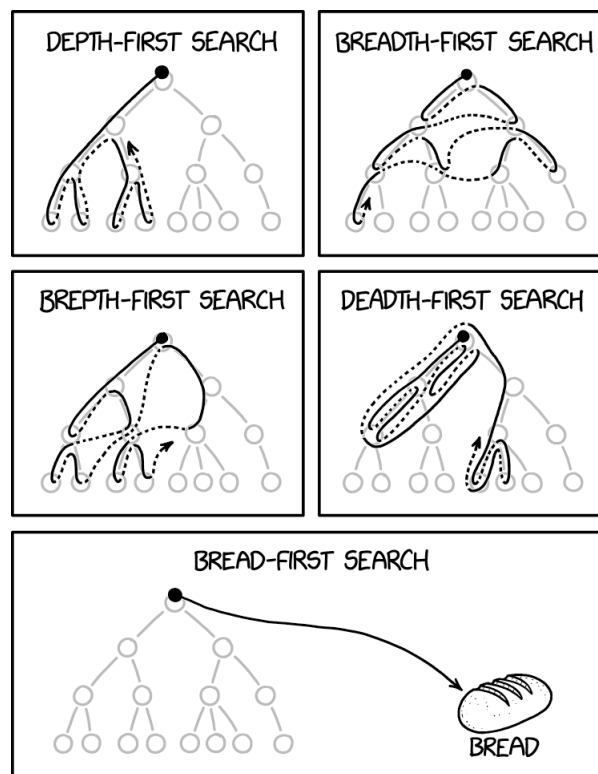


Chapitre 3

Union-Find

(Programme de khôlles)



Groupes A, B & C (CCINP et Mines-Telecom)

1. Rappeler le principe d'une structure Union-Find, les fonctions nécessaires
2. (Pour les révisions de concours) Rappeler l'utilité d'une structure Union-Find, quel/quels algorithme/algorithmes au programme peut/peuvent l'utiliser?
3. Implémentation basique sans optimisation : savoir la coder en OCAML et en C, savoir les complexités de chaque fonction (demos?) et savoir les critiquer (du moins voir où des améliorations peuvent être faites).
4. Optimisation première - union pondérée : savoir l'expliquer et l'implémenter dans les deux langages.
5. Optimisation seconde - compression des chemins : savoir l'expliquer et l'implémenter dans les deux langages.
6. Savoir montrer la différence entre avec et sans optimisation sur un exemple.
7. Proposition : La complexité amortie des deux opérations est en $\mathcal{O}(\alpha(n))$ où α est l'inverse de la fonction d'Ackermann.

Groupes B & C (Mines, Centrale, X)

8. Théorème : Grâce à l'optimisation première, une classe de rang k possède des chemins de longueur maximale k et au moins 2^k éléments (démonstration).
9. Corollaire : Grâce à l'optimisation première, on obtient des opérations **find** et **union** en $\mathcal{O}(\log(n))$ (démonstration).

Groupe C (ENS)

10. Fonction d'Ackermann (définition).

⚠ Ce programme paraît court mais est extrêmement utile et demandé, vous ne devez ABSOLUMENT PAS hésiter sur l'implémentation de la structure avec les deux optimisations!!