

Diviser pour régner

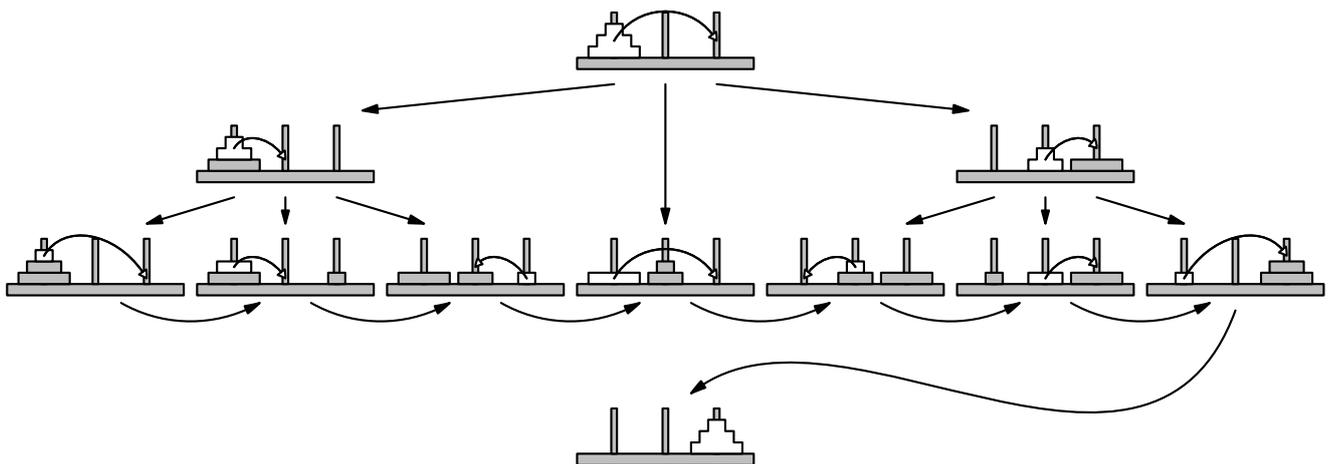
Diviser pour régner

On dit qu'un algorithme suit le principe de **diviser pour régner** (*divide and conquer*) s'il fonctionne selon le principe suivant :

1. **Phase *diviser***: décomposition du problème à résoudre en problèmes plus simples (souvent en divisant la taille des données par 2) ;
2. **Phase *régner***: résolutions des sous-problèmes, souvent triviaux ;
3. **Phase *combiner***: reconstruction de la solution globale à partir des solutions des sous-problèmes.

La nature même de cette approche se prête très bien à la récursivité. D'ailleurs plusieurs algorithmes récursifs vu cette année suivent cette approche. C'est le cas de la somme des nœuds dans un arbre binaire. On fait la somme des deux sous-arbres et on ajoute la valeur de la racine.

La résolution des tours de Hanoï est un exemple canonique de cette approche :

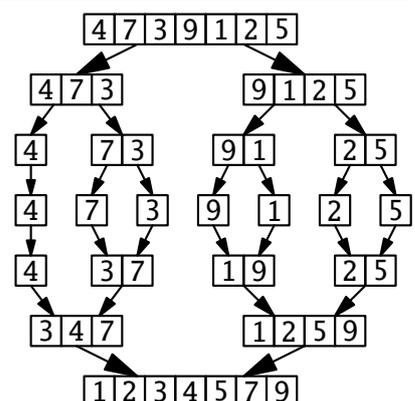


Pour déplacer une pile en entier, on commence par déplacer une pile de taille inférieure. On continue à diviser le problème jusqu'à arriver à des déplacements de disque, que l'on peut faire facilement. La phase de combinaison correspond juste à faire les opérations dans le bon ordre.

Tri fusion

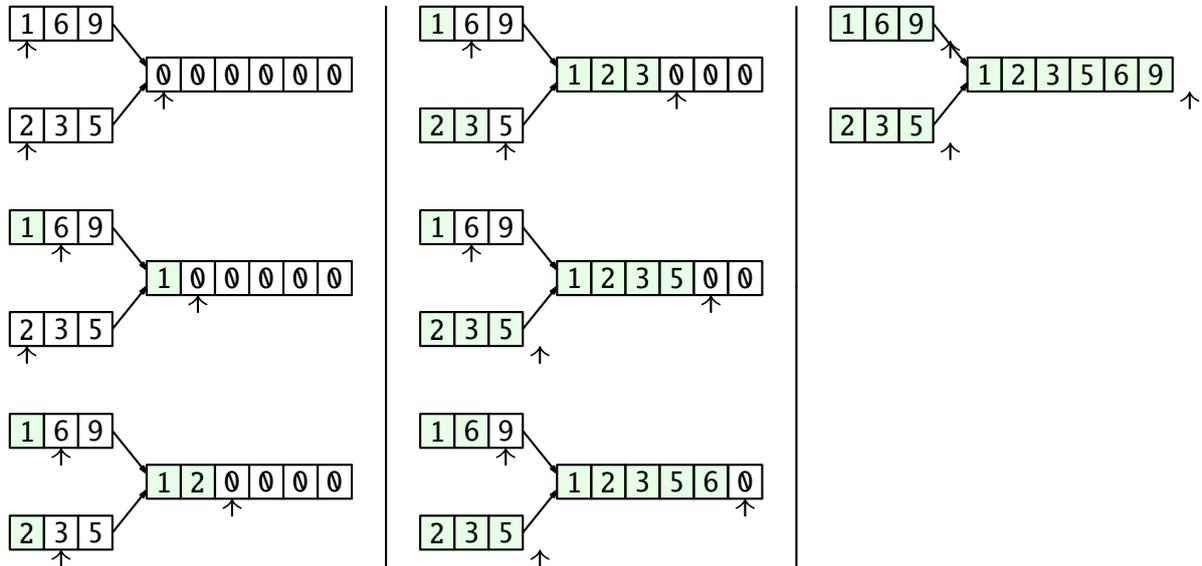
Le **tri fusion** est un autre exemple classique d'algorithme diviser pour régner. Il consiste à découper le tableau en deux, jusqu'à arriver à des tableaux ne contenant qu'un seul élément, puis à les fusionner en formant des tableaux triés. En effet, fusionner deux tableaux triés est plus rapide que de trier directement le tableau complet.

La complexité de cet algorithme est en $O(n \log_2(n))$, ce qui est optimal pour un algorithme de tri. Les autres algorithmes que nous avons vu sont en $O(n^2)$.

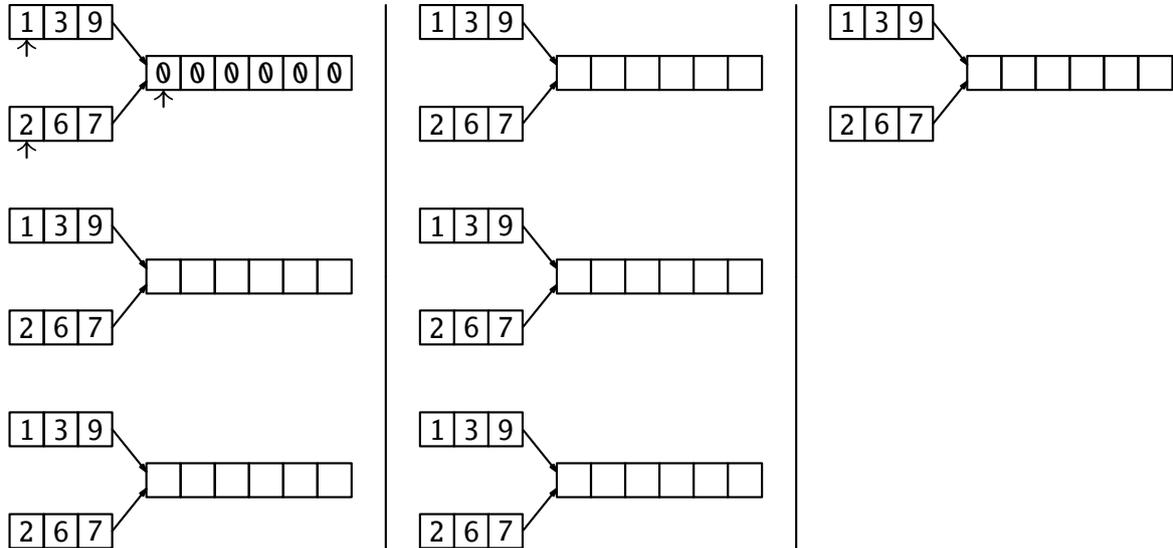


Le principe de la fusion de deux tableaux est le suivant :

- On crée un nouveau tableau dont la longueur est la somme de celles des deux autres.
- On se place au début de chacun des deux tableaux.
- On met le plus petit des deux éléments au début du nouveau tableau et on avance dans le tableau où on a pris le minimum.
- On continue jusqu'à avoir parcouru un tableau en entier. Il ne reste plus qu'à copier le reste de l'autre tableau.



EXERCICE 1 : Réaliser la fusion de ces deux tableaux en indiquant la position des indices.



EXERCICE 2 : On considère deux tableaux triés de longueur n $tab1$ et $tab2$.

- 1) Combien de tests entre des éléments de $tab1$ et $tab2$ faut-il au minimum pour les fusionner?
- 2) Combien de tests entre des éléments de $tab1$ et $tab2$ faut-il au maximum pour les fusionner?
- 3) Combien de tests faut-il pour trier un tableau de $2n$ éléments avec un algorithme comme le tri à bulles?