2 minutes – Algo

Notation

1. Prouver que f(n) = … (une complexité)
   ➔ Calculer la limite vers + l'infini puis cf. refcard
2. Une variable dans f(n) peut être nulle pour que l'égalité du 1. soit vrai.
   ➔ Avec un peu de réflexion… ça se trouve. Dans le doute, la variable c

Savoir faire des calculs de complexités

Savoir faire des rotations dans un arbre binaire de recherche

Arbre n-aire

On considère ici des arbres n-aires, c'est-à-dire des arbres dont chaque nœud possède soit n fils soit aucun fils.

Nombre max de feuilles pour un arbre n-aire de hauteur h : n^h

Nombre max de nœuds pour un arbre n-aire de hauteur h : $\sum_{i=0}^{h} n^i = \frac{n^{h+1}-1}{n-1}$

Hauteur minimal h pour m nœuds : s'aider de l'équation dessus pour trouver h

Hauteur max pour m nœuds pour un arbre n-aire : $\left\lceil \frac{m-1}{n} \right\rceil$

Algo de tri

Tri stable

Tri par insertion $O(n^2)$ Tri fusion θ(nlog(n)) Tri par dénombrement θ (n)

Tri instable

Tri rapide $O(n^2)$ Tri par sélection θ (n^2) Tri par tas O(nlog(n)) Tri introspectif O(nlogn)

Indiquez une façon de rendre stable n'importe quel algorithme de tri. Quelle complexité (en temps et espace) cette modification ajoute-t-elle à un algorithme de tri lorsqu'il fauttrier n valeurs ?

Il suffit de modifier les clefs des valeurs à trier pour y inclure leur indice d'origine. La fonction de comparaison doit alors être adaptée pour comparer ces indices lorsque les parties originales des clefs sont identiques.
Ce changement demande Q(n) mémoire supplémentaire (on ne peut donc plus prétendre que le tri soit en place, car l'occupation mémoire dépend de n), en revanche il ne modifie pas la complexité du temps d'exécution.

Misc

Insertion dans un tas : O(log(n))

Complexité de rechercher la k de plus grande valeur dans un arbre rouge et noir de n nœuds : O(n)

Il y a 2^8 = 256 entiers sur 8 bits. 70 possibilités d'entiers équitables sur 8 bits. Il y a donc 14 entiers convenables sur 8 bits