Preuve d'algorithme
Examen d'algorithmique 2009

FMdKdD

8 janvier 2010

Démonstration. Soit i l'intervalle donné en argument. Montrons que la méthode `rechercher` est correcte, c'est à dire qu'elle renvoie soit un intervalle qui recoupe i, soit `null` car aucun intervalle dans l'arbre ne recoupe i.

En regardant le code de la méthode, on voit tout de suite que x parcoure l'arbre en partant de la racine. À tout moment, il est aisé de vérifier que x est soit `null` soit un nœud de l'arbre. La boucle `while` ne cesse de s'éxécuter que si l'une des deux conditions suivantes est vérifiée :

- x est `null`
- x recoupe i

Comme ces conditions sont mutuellement exclusives, si x n'est pas `null`, c'est qu'il recoupe i. Si la méthode `rechercher` renvoie un intervalle, il appartient à l'arbre et recoupe i.

Il reste à montrer que la méthode renvoie `null` si et seulement s'il n'y a pas d'intervalle qui recoupe i dans l'arbre. Notons $i.a$ et $i.b$ le début et la fin d'un intervalle i, respectivement.

L'algorithme suppose qu'il existe au moins un intervalle k dans l'arbre qui recoupe i. Montrons qu'il renvoie k s'il existe, ou `null` sinon.

Pour que k recoupe i, il faut vérifier : $k.a \leqslant i.b$ et $i.a \leqslant k.b$.

Lorsqu'un tour de boucle commence, x n'est pas `null` et ne recoupe pas i. Il faut choisir parmi les deux fils de x celui qui contient k (ou éviter celui qui ne le contient pas) à l'aide du test $i.a \leqslant mG$.

1. Si $i.a > mG$, où mG est la plus grande fin d'intervalle trouvée dans le sous arbre gauche G de x, alors $i.a > y.b, \forall y \in G$ et aucun y de G ne pourra recouper i. L'intervalle k, s'il existe, se trouve donc dans le sous arbre droit D de x.
2. Si $i.a \leqslant mG$, est-il possible que $k \in D$ et qu'aucun $y \in G$ ne recoupe i ? Autrement dit, doit-on forcément aller vers le sous arbre gauche si $i.a \leqslant mG$? Supposons $k \in D$ et $\forall y \in G$, y ne recoupe pas i. On a, $\forall y \in G$:

$$i.a > y.b \text{ ou } i.b < y.a$$
$$y.a \leqslant x.a$$

Et, comme $k \in D$:

$$k.a \leqslant i.b \text{ et } i.a \leqslant k.b$$
$$x.a \leqslant k.a$$

Si $i.b < y.a$, alors

$$y.a \leqslant x.a \leqslant k.a$$
$$i.b < y.a \leqslant x.a \leqslant k.a \leqslant i.b$$
$$i.b < i.b$$

Absurde ! Sinon, c'est que $i.a > y.b$, $\forall y \in G$, donc $i.a > mG$. Mais rappelons nous que $i.a \leqslant mG$. Absurde. Donc on ne peut pas avoir $k \in D$ et aucun intervalle qui recoupe i dans G.

Il reste trois possibilités :

- soit $k \in D$ et il existe alors un $k' \in G$ tel que k' recoupe i, et on continue avec k' à la place de k,
- soit $k \in G$,
- soit k n'existe pas, et quel que soit le chemin pris, on finira par tomber sur une feuille et on renverra `null`.

Dans ces trois cas, il faut aller vers le sous arbre gauche.

Par conséquent, se diriger vers G si $i.a \leqslant mG$ nous rapproche de k, et s'il existe l'algorithme le trouve ainsi. □

Remarque. Notons que l'algorithme décrit n'est pas optimal : il continue de parcourir l'arbre A quand on peut être sûr que l'on ne trouvera pas de solution. Par exemple, si $i.a > y.b, \forall y \in A$, alors $i.a$ est plus grand que le `maxFin` de la racine, et l'algorithme parcoure tout de même l'arbre alors qu'aucun $y \in A$ ne peut satisfaire $i.a \leqslant y.b$.